PCT

WORLD INTELLECTUAL PROPERTY ORGANIZATION
International Bureau

INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(51) International Patent Classification 3 : G06F 3/02	A1	(11) International Publication Number: WO 85/ 01596 (43) International Publication Date: 11 April 1985 (11.04.85)

(21) International Application Number: PCT/AU84/00193

(22) International Filing Date: 4 October 1984 (04.10.84)

(31) Priority Application Number: PG 1692

(32) Priority Date: 4 October 1983 (04.10.83)

(33) Priority Country: AU

(71) Applicant *(for all designated States except US)*: KEY-CORP PTY. LIMITED [AU/AU]; 22 Lambton Road, Broadmeadow, NSW 2292 (AU).

~~(72) Inventor; and~~
(75) Inventor/Applicant *(for US only)* : WOOD, John, William [AU/AU]; 27 Ilford Avenue, Buttaba, NSW 2300 (AU).

(74) Agent: TAYLOR, Paul, Robert; Arthur S. Cave & Co., Gold Fields House, 1 Alfred Street, Sydney, NSW 2000 (AU).

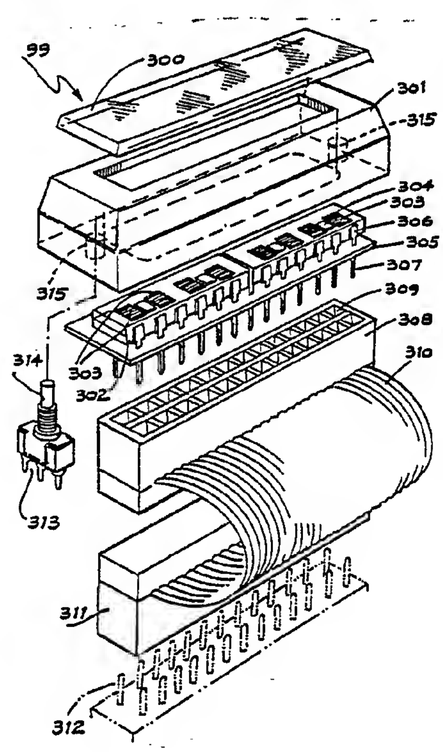
(81) Designated States: BE (European patent), BG, BR, CH (European patent), DE (European patent), DK, FR (European patent), GB (European patent), HU, JP, KP, KR, NL (European patent), NO, RO, SE (European patent), SU, US.

Published
With international search report.

(54) Title: TACTILE FUNCTION KEY

(57) Abstract

A positive feedback tactile function key (99) comprising a single or multicharacter non-specific display (102), means for activating said display, and means for interfacing said function key (99) with an electronic system. Also disclosed is a system for interfacing a device that uses electronic means to label keys so that use of the operator interface advantages of such a device can be gained with programs that were not designated for use with it.



FOR THE PURPOSES OF INFORMATION ONLY

Codes used to identify States party to the PCT on the front pages of pamphlets publishing international applications under the PCT.

AT	Austria	GA	Gabon	MR	Mauritania
AU	Australia	GB	United Kingdom	MW	Malawi
BB	Barbados	HU	Hungary	NL	Netherlands
BE	Belgium	IT	Italy	NO	Norway
BG	Bulgaria	JP	Japan	RO	Romania
BR	Brazil	KP	Democratic People's Republic of Korea	SD	Sudan
CF	Central African Republic			SE	Sweden
CG	Congo	KR	Republic of Korea	SN	Senegal
CH	Switzerland	LI	Liechtenstein	SU	Soviet Union
CM	Cameroon	LK	Sri Lanka	TD	Chad
DE	Germany, Federal Republic of	LU	Luxembourg	TG	Togo
DK	Denmark	MC	Monaco	US	United States of America
FI	Finland	MG	Madagascar		
FR	France	ML	Mali		

TACTILE FUNCTION KEY

The present invention relates to improvements in function keys for use with electronic systems.

Function keys were developed as an aid to the operator and are used to control the processing to be done on an electronic system. At present there are three types of Function keys:

(1) The control (CTRL) and escape (ESC) keys which operate in conjunction with the standard keys e.g. typing CTRL quote C unquote on many systems will have the effect of aborting the program being processed.

(2) Particular Function keys which are identified as such by having the particular function printed on them e.g. Edit.

(3) Soft keys. These keys are simply identified as F1, F2, F3 etc. and are used by the operator to select the function or option desired.

It will be apparent to those skilled in the art that Function keys of the type (1) and (2) above are strictly limited in their commands whereas Function keys of the type (3) offer a range of commands or options. However type (3) Function keys suffer from a considerable disadvantage in that they are only identified as F1, F2, F3 etc. The user must refer to a program code chart to identify each Function key; alternatively the user may memorise the required sequence from familiarity with that particular program. However as each program is controlled by different Function keys and related sequence thereof it is usually necessary to consult the program code chart each

time.

In order to partially overcome this disadvantage the user can physically affix a label to each Function key indicating the name of each function or on some computers the Function keys are identified on the terminal screen.

The disadvantage with the first method of identifying the Function keys is obvious and whilst the second method is preferred it does limit the available space on the terminal screen, and for this reason the identification of the Function keys is often deleted from the screen by the operator.

Recent developments are the Apricot micro-computer which incorporates in its keyboard a separate microscreen placed adjacent to six touch sensitive Function keys comprising a two line Liquid Cyrstal Display, which can be used to label the six Function keys with functions unique to the program currently being run and the Hewlett Packard HP150 computer which incorporates a touch sensitive screen.

Whilst going some way towards overcoming the above disadvantages the Apricot microscreen is still limited to and dependant on the program being run on the computer; it displays a function definition for an adjacent key rather than each key having an independant display, it is only operable with touch sensitive function keys that are not operator preferred. In addition the microscreen is not specifically designed as a funtion key tool as it also operates as a clock, calendar, calculator or auxillary terminal screen. Moreover the microscreen and the

associated touch sensitive Function keys are specifically designed for and operated by Apricot software and are not independant of any particular electronic system.

Hewlett Packard has released a microcomputer with a "touch sensitive" screen. Areas of the screen can be defined to represent a particular function, and upon pointing at this region ("touching the screen") the operator breaks a series of infra red beams and the computer interprets this as selection of that function. This method is not favoured because it takes the operator's hands away from the keyboard, and makes the operator reach to the screen. This action of reaching to the screen is an uncomfortable procedure for the operator.

All software used with the touch sensitive screen must be designed specifically for use with it, and it is almost impossible for the casual programmer to generate his own programs to make use of this facility.

The present invention seeks to alleviate these disadvantages by providing a set of self-identifying Function keys, which are independently programmable and which can interface with any computer or dumb electronic system.

It has been discovered that it is possible on a tactile key to have a multicharacter non-specific display without losing the compact dimensions of a conventional key. It is therefore possible to incorporate the Function key of the present invention in ordinary keyboards.

In one broad aspect of the present invention there is

provided a tactile function key comprising a single or multicharacter non-specific display, means for activating said display, and means for interfacing said function key with an electronic system.

In a further aspect of the present invention there is provided a peripheral keypad comprising one or more tactile function keys, each function key comprising a single or multicharacter non-specific display, means for activating said display, and means for interfacing said function key with an electronic system.

In a still further aspect of the present invention there is provided a keyboard that includes a keyset comprising one or more tactile function keys, each function key comprising a single or multicharacter non-specific display means for activating said display, and means for interfacing said keyboard to a host computer.

In another aspect of the present invention there is provided a keypad or keyset comprised of one or more tactile function keys, each key comprising a single or multicharacter non-specific display, means for activating said display, and means for interfacing said function key with a computer system so that while running a main level program, each function key can be used to "ghost" a set of signals to the host computer as if they came from the computer's keyboard keys, and both the function definitions displayed on each function key, and the characters "ghosted" when each function key is depressed can be changed for each function key without affecting the

execution of the main level program.

In another aspect of the present invention there is provided a keypad linked to, or keyset incorporated in, a computer keyboard, said keypad or keyset comprising one or more tactile function keys, each key comprising a single or multicharacter non-specific display and having means to monitor the conventional keystrokes being input from the computer's keyboard, and on entry of the set of characters defined as input upon depression of any one of the function keys, the function definitions displayed on each function key, and the keystrokes to be "ghosted" upon the subsequent depression of a function key may be changed in an identical manner as if that particular function key were depressed instead. This process will be referred to as "function key equivalence checking".

In a still further aspect of the present invention there is provided a keyset comprising one or more tactile function keys, each function comprising a single or multicharacter non-specific display, means for activating said display, means for interfacing said function keys with a "dumb" electronic system, including program means to control the operation of said "dumb" electronic system.

Glossary of Terms used in this Specification

"intelligent"	electronic system: eg computer
"dumb"	electronic system: eg cash register, video, microwave oven, photocopier
RAM	random access memory
ROM	read only memory

EE ROM	electronically erasible programmable read only memory
"ghost writing"	by depressing one of the function keys of the present invention set of signals is sent to program open to input as if they were operated by conventional means
"touch sensitive"	a flat surface which sends a signal when pressed, in contrast to a normal key which stands proud out of a surface and provides a positive tactile feedback upon key depression.

The Function keys of the present invention are non-specific, that is each key can represent an almost infinite number of functions or options. These functions or options are specified by commands from within the program being run itself; the electronic system software or independantly of either by printed circuit card or microprocessor co-operating with the Function keys to act upon a particular "intelligent" or "dumb" electronic system.

Once specified the Function key will visually display symbols representative of the function or option in single or multicharacter form on its face.

The displays employed in the production unit will depend upon comsumer preference, economics, physical restrictions, and the state of display technology. It is probable that a range of units will be marketed, with selected different types of displays available.

Examples of some types of displays are as follows:

(a) Dot Matrix Light Emitting Diode Displays

Dot Matrix Light Emitting Diodes have the following advantages:

1) Very good quality character definition.

2) Display units such as the HDSP2000 are very compact.

3) Durable and fairly rugged.

4) Economic for large scale production.

5) With the driving die included in the display surface such as on the HDSP2000, there are relatively few connector pins.

6) Displays such as the HDSP2000 have been well proven over time.

The major disadvantages are:

1) Non-economic for small scale production.

2) Use a lot of power. It is probable that an independant power supply, separate to that of the main computer, would be necessary to run the displays. This would lead to difficulties in implementation of the unit within the computer keyboard.

3) Heat Generation. The displays generate quite a deal of heat. This problem is solved in the prototype by the use of aluminium key housings which act as a heat sink. A heat sink would be necessary for the displays used in a production model.

(b) Dot Matrix Liquid Crystal Displays

Advantages:

1) No heat generation.

2) Little power required. All power necessary could be

drawn through the computer keyboard plug.

3) Economic for production.

4) Good quality character definition.

5) Using chip on glass technology the number of connections to the display can be minimised, however as this technology is still fairly young the same degree of refinement as in the Light Emitting Diode displays has not been reached and thus a physically larger display for the same character size results.

6) By incorporating the drivers for the display on a printed circuit board immediately below the display, all necessary connections can be made using a zebra strip. If this method is used correctly, a very compact fairly rugged unit eventuates.

Disadvantages:

1) Generally not as rugged as Light Emitting Diode Displays.

2) Chip on glass technology is fairly new.

3) There is a tradeoff between the viewing angle for the display and the amount of multiplexing that is done to the data for the displays.

The current state of liquid crystal technology allows the production of compact displays using either Chip on Glass technology or the integrated unit described above. The Conventional liquid crystal displays are plentiful, and drivers such as the Phillips PCF8576 will perform the driving functions required. Two of these drivers will drive an 8 character 8x5 dot matrix Liquid Crystal display

where the bottom line is used to underline characters. The maximum multiplexing ratio for these chips is 4:a. This allows a wide viewing angle with good clarity, and allows design of a near symmetrical unit with one die driving the top four lines, and the other die the bottom four.

(c) Starburst Displays

The advantage of the starburst display is that it has less segments that must be controlled, and hence the whole circuit becomes simpler. With displays, it is often the "driver" that controls the configuration of the display that is the most expensive. If there are less segments on a display then less drivers are necessary, or the drivers can be replaced with simpler cheaper ones. Thus starburst displays are much cheaper than their dot matrix counterparts. The display units can also be made more compact, with less connections to individual segments being necessary.

The starburst format is available in both liquid crystal and light emitting diode displays. The obvious disadvantage of the starburst display is that it is restricted in the number of different characters it can display and does not produce the same quality of character definition as does dot matrix displays.

The present invention will be more fully understood by reference to the preferred embodiment as illustrated in the accompanying drawings, in which:

Fig. 1 is an exploded perspective view of a Function key of the present invention incorporating a liquid crystal

display;

Fig. 2 is a section view of the Function key of Fig. 1;

Fig. 3 is an exploded perspective view of a Function key of the present invention incorporating a light emitting diode display;

Fig. 4 is a section view of the Function key of Fig. 3;

Fig. 5 is a schematic view of a keyset of the present invention linked to a computer;

Fig. 6 is a schematic view of a keyset inset in a keyboard;

Figs. 7a and 7b is a circuit diagram of the logic behind the keyset of Fig. 5 and its interface with a computer;

Fig. 8 is a schematic flowchart of the logic behind a write to keyset character RAM;

Fig. 9 is a schematic flowchart of the logic behind a read instruction on depression of a Function key;

Fig. 10 shows a series of wave forms generated by the timing circuitry of the present invention;

Figs. 11a and 11b is a circuit diagram of the logic behind a keypad of Fig. 5 interfacing through an IRS232;

Figs. 12a and 12b is a circuit diagram of the logic behind the keypad of Fig. 5 intercepting the keyboard cable;

Figs. 13a and 13b is a circuit diagram of the logic behind the keyset of Fig. 6 intercepting with the keyboard cable;

Fig. 14 shows a system of pointers for use in "ghost" writing.

The tactile Function key of the present invention comprises a single or multicharacter non-specific display as shown in Figs. 1 and 2. The tactile Function key 99 has a keytop 100 which is depressed by the operator. Beneath this keytop 100 lies a protective polarised glass plate 101 which protects from damage the liquid crystal display (LCD) assembly 102. Two conductive rubber "zebra" strips 103 connect the LCD assembly to the printed circuit board (P.C.B.) 104. On this P.C.B. board are located two dies 105 each die being pin connected via selected pins 106 to one end of a flexible printed cable strip (P.C.S.) 107, the other end of which is detachable from connectors 108 located on a base plate 109.

The keytop 100 is supported on zebra strips 103 and if needed by flanges at the ends of board 104 and locates with base plate 109 by means of a moulded plastic insert 110.

In the centre of the base plate 109 lies a conventional switch assembly 111 comprising a mounting 112 and a plunger 113.

The construction of the function key 99 is designed so that it will assemble or disassemble in three steps. To assemble the key all of the display components are put in place and the clip 115 is clicked into co-operating groove 114. The flexible cables are put into their slots and then the key is then placed on the plunger 113 and clicked into co-operating groove 116 using clip 117. Clip 117 maintains the display 102 in contact with the zebra strips 103 at all times. The reverse procedure is followed for disassembly.

Clip 116 is designed to be opened with a screwdriver or similar instrument for disassembly.

Descriptions for LED

In Figs. 3 and 4 is shown an alternative display namely a Light Emmitting Diode display (LED). The tactile function key 99 comprises a protective glass top 300, a keytop 301 made of aluminium and acting as a heat sink as described above. At each end of the keytop 301 are slots 315 to receive, in spigot fashion, the plunger 314 on switch 313. The display on board 302 comprises eight LED's 303 interspaced by dies 304 and having a protective glass 302', on top of the displays. The displays are connected to a board 305 having 13 pairs of pins 307 to connect in spigot fashion with holes 309 on board 308. The pins 306 are connected to pins 307 via conductive paths. The middle pair of pins 307 not being used. A flexible P.C.S. 310 flexes between board 308 and an identical board 311. Board 311 in similar spigot fashion connects with pins 312 on main printed circuit board (not shown).

In Fig. 5 the Function keys of the present invention are incorporated in a keypad 90. This keypad is interfaced with a computer 70 having a screen 71 and a key board 80, having conventional keys 81.

In Fig. 6 there is shown a keyboard 180 having conventional keys 181 and ten tactile Function keys 99 incorporated therein.

In Figs. 7a and 7b are shown the circuitry and operation of a keyset interfaced with an IBM personal

computer. The components U1 - U29 are conventional components, their code numbers being listed on the drawings. Each component has conventionally numbered pins associated therewith.

For brevity and simplicity only, Figs. 8 and 9 respectively illustrate schematically a write/read statement to/from a single function key.

In these drawings the following components are given symbols:

B1 - B6 are buffers

C - Microprocessor/computer

D - driver

E1 - E3 are encoders/decoder,

K - Function key multicharacter

M - multiplexer

P - power supply

R - character RAM

S - switch

T - Timing circuitry

200 - IRQ2 request

201 - Data Bus

202 - Address Bus

203 - IO request

204 - write/read

205 - keyboard read

206 - Display write

207 - Row data parallel

In Figs. 11, 12 and 13 as in Figs. 7a and 7b

convention code numbering is given to the various components.

Keypad Without Microprocessor - Circuit Description

There are many different ways in which a keypad comprising one or more tactile function keys where each function key comprises a single or multicharacter non-specific display can be interfaced to a computer. The most favourable methods are via a RS232 interface (or other standard interfaces), via an expansion slot on the computer, or by diverting the computer keyboard cable to plug into the keypad, interpreting the signals received from the computer keyboard and then sending this information, and any information from the function keys along the keypad's cable to the regular keyboard socket on the host computer.

Unit that Operates Via an Expansion Slot

The unit designed to interface to the host computer via an expansion slot has the following characteristics:

1) A RAM chip is used to store the function descriptions currently being displayed on the function keys.

2) A ROM chip, or a EE ROM chip, or a ROM chip with an additional optional selectable RAM is used to store the information necessary to display each character of the character set currently available for display on the function keys.

Normally the circuit continually cycles, controlled by a timing circuit, taking the data to be displayed from the RAM, encoding it with the information in the ROM (or EE ROM

or selectable RAM), and serialising it for the display.

When it is required to change the information in the character RAM (See Fig. 8), the following procedure is followed:

To write display information to character RAM an IO write 204 is done to the appropriate RAM location. The decoding circuitry E3 enables buffers B1 and B3, and sets the direction of buffer B1 to allow data transfer in the direction of the RAM, (arrowed in Fig. 7), and enable buffer B5 and disables buffer B6 to give control of the address bus 202 to the computer (arrowed in Fig. 7). The write pin of the RAM is strobed and the data deposited on it. A similar procedure is followed if an EE ROM or ROM with additional optional selectable RAM is being used, and a new character set is to be dumped to this memory area.

When a key is depressed, the following procedure occurs:

As shown in Fig. 9, when a key is depressed on the keypad, this is detected by the Keyboard Encoder E1 and an interrupt request 200 is issued to the computer C. When the computer C responds to this interrupt by performing an IO read (203) at the keypad address, buffers B1 and B2 are enabled, the direction of buffer B1 is set toward the computer, buffer B3 is disabled and the keyboard encoder E1 places the code for the key depressed, onto the data bus (201). The rest of the circuitry is in its normal state (buffer B5 disabled) and the contents of the RAM R are continuously strobed into display via the character

generator ROM (or EE ROM or ROM with optional selectable RAM) and multiplexor M.

A full and detailed description is given below for a unit that will communicate with an IBM PC via a slot in the expansion bus.

Example: Keypad to Interface an IBM PC via an Expansion Bus

The keypad consists basically of a four by two matrix keyboard which outputs an interrupt on a keypress, the data being made available at the selected output address, and eight "8" character displays mounted within the key-tops and driven from RAM which can be loaded from the IBM.

Display Controller Circuit

The controller circuit (Fig. 7a and Fig. 7b) is designed to accept standard ASCII characters from the PC for storage in the local 128 x 8 ram (68A10). The RAM can be assigned to one of four blocks, starting at 400H, 500H, 600H and 700H and for a 64 character display, will utilise the lower 64 locations. e.g. 400H - 43FH. The ASCII data may be loaded sequentially with new data, or individual characters may be altered by writing the new data to the corresponding location in RAM.

The last character in the bottom right hand key is loaded with data from the first RAM location and the first character of the top left hand key corresponds to the last RAM location. After data has been loaded into the display RAM, the local scanning circuitry controls the decoding of the ASCII characters, the display data loading and the

column select function.

The address bus and control signals are buffered by U5 and U2 and fed to the decoding circuitry which allows the appropriate buffers to be turned on and off to facilitate taking the local scanning circuitry off the bus whenever data is required to be loaded into the display ram.

Normally, buffers U1, U11, U25 and U26 are disabled and U10 is enabled. The system clock is divided down to give approx. 500KHz and fed to U9 which drives the display scanning circuitry. U15 cycles through the 64 locations used in the RAM (U16) and places the previously stored ASCII data onto the address inputs of the 2716 EPROM (U21). U21 is used as a 128 character ASCII to 5x7 dot matrix decoder. The lower 5 data outputs of U21 contain the data required for one row of the one character determined by the data fed into the seven most significant address inputs from the data outputs of U16. The other six rows are presented to the output by cycling the lower three address inputs via U9, while the data from U16 is held constant.

The five column outputs of the EPROM (U21), are gated to the Data Input of the display via the 74151 multiplexer (U20). Strobing of the display is accomplished via the 74197 (U9), 74393 (U15) and 7490 (U19) counter string. U9 is connected as a "divide by 8" counter that sequentially selects the seven rows within U21 and also enables seven clock cycles to be gated to the clock input of the display.

Referring to the attached timing diagram 10, waveform

W is the 500KHz clock applied to pin 8 of U9 and waveform X is the output of pin 2 of 7404 (U30) which is the inverted Qd output of U9. Waveform Y is the output of the 7404 (u29) pin 6 which is the ANDED output of 2Qc and 2Qd of U15.

The 74393 (U15) is configured as a divide by 256 counter connected so that the six lowest order outputs select each of the 64 ASCII characters within U16. The three highest order outputs determine the relationship between load time and column on time. When the outputs 2Qc and 2Qd are high, the display clock is enabled, and the circuit scans the 64 characters in the U16, serialises the column data by counting through each of the seven rows of the 2716 and gating the appropriate column of the display. During the three counts when the two most significant outputs of U15 are not both high, the display clock is disabled, ensuring stable data in the display buffer, and the column data is displayed via U17 and darlington drivers as shown by waveform Z.

When the data on the display is to be changed, the combination of a valid address and a write pulse (CS.IOW) disables U10, removing the counter U15 from the address inputs of the ram, and U11 is enabled to place the address bus in its place. Buffer U1 is enabled and directed "IN" and data is written into U16 via the buffer U25 which is also enabled CS.IOW. Once the CS or the IOW is removed, the buffers resort to their original states and the display is once again cycled.

Keypad Operation

The keypad 90 is interfaced with, in this illustration, an IBM computer 70 using a 74C922 keyboard encoder U27 which contains all the logic to fully encode an array of SPST switches. The keypad is scanned asynchronously and upon a key depression being detected, a Data Available signal is forced high on pin 12 after a preset debounce period, and returns to a low level after the key is released. This signal is latched by U28 which sets IRQ2 on the IBM computer 70.

Keyboards and Keypads Utilizing a Microprocessor

Utilizing a microprocessor to control the function keys increases the flexibility of the interface methods that can be used with them. The keyboard and keypad can be developed to link to any available bi-directional input/output port or pair of opposed uni-directional ports on a computer. As almost every computer or computer terminal has such a port it is obvious that a keypad and/or keyboard can be developed using this technique for almost any computer or terminal. A condition placed on the keyboard provides that the keyboard plug through which the unit is to communicate with the computer must have an appropriate bi-directional line or have two separate unidirectional lines running in opposite directions between the keyboard and the computer. If no such line is available, this problem can be overcome by using a bi-directional port such as an RS232 port for all keyboard communication. To achieve this, the computer must be able

to allow data input through this port in the same manner as it would through the keyboard plug. As most computers are designed to allow outside communication ie. through a modem, this will not often present a problem. A design for a keyboard utilizing a RS232 interface is not included however it will be obvious to one skilled in the art that the technique as shown in Fig. 11 for the keypad is equally applicable to the keyboard as shown in Fig. 13.

In order to interface such a keyboard or keypad or a keyset of function keys into a "dumb electronic device" it may be convenient to exchange the RAM memory for ROM memory which has been pre-programmed with the required data. The output from the microprocessor would be tailored to that required by the device however note that with the appropriate interface circuitry the methods of "ghosting" and/or "function key equivalence checking" can be adapted so that the key can be used with devices designed without consideration for their use. The objects "ghosted" may not necessarily be character codes but alternatively may be millivolt signals etc.

Full circuit descriptions follow for three units that utilize a microprocessor control the use of the function keys, each having a different interface method. No circuit description is included for a unit to interface to a "dumb electronic device" but with the information in Figs. 11 - 13 it would be obvious to one skilled in the art that all that is required is to develop an interface circuit to suit the device being used.

The three units described are:

(i) IBM Keyboard Substitutes - this unit performs exactly as a standard IBM unit except for the added Function keys.

(ii) RS232 Peripheral - this device has only the Function keys and communicates with any computor via an EIA standard RS-232C front.

(iii) Interface unit - this device is connected electrically between the IBM-PC and an IMB-PC keyboard. It has only the Function keys fitted.

All three services can be described as consisting of the following functional sections:

1. Microcontroller
2. Memory
3. ZIF Displays
4. Keyboard Scanner
5. Communications interface

The first three sections are identical in all three versions. Item 4 is identical in versions (ii) and (iii).

1. Microcontroller

The microcontroller is a commercially available Intel 8051 single chip microprocessor. Detailed description of its operation is complex and can be found in published technical data manuals. It can however be generally described as a microprocessor operating with a clock frequency of 4 MHz and interfacing into our application as four 8 bit bi-directional parts. Part 0 is used as a multiplexed part for the lower memory address byte and an 8 bit data path. The lower 5 bits of part 2 constitute the

upper address for the external memory. The upper two bits 6, 7 are used for serial communcations with the display control devices as serial data and serial clock respectively. Parts 1 and 3 are used for keyboard scanning and communications.

2. Memory

Although the Intel 8051 has internal memory this application calls for more capacity so two Jedec standard 28 pin memory sockets are provided. These sockets can hold either Ram or Eprom devices of capacity 8k bytes each as required. In order to interface the memory with the 8051 an LS373 octal latch must be provided. This latch when strobed by the ALE control line latches the lower address byte to the memory sockets and releases PORE 0 to accept or output data to the memory devices. The reading and writing to and from RAM in these memory sockets is enable by the RD and WR lines which are special functions of Part 3 bits 7 and 6 respectively.

3. Function Key Displays (190)

The Function key displays basically comprise of liquid crystal dot matrix panels each 40 dots wide by 8 dots high enabling a wide choice of characters. These displays are driven by intelligent LCD display controllers - PCF8576. Each of these drivers has its own individual address which is coded by the A_o, A_1, A_2 and SAO fins. When coupled with software control addressing each device can respond accordingly. Communications with these devices are carried out along the SDA (serial data line) and the SCL

(clock logic line) and is the only necessary interface with the microcontroller. Details on the serial protocol can be found in manufacturer's data sheets. Bits 6 and 7 on I/O part 2 are used by the microcontroller to control the SDA and SCL lines respectively.

4. Keyboard Scanner

(i) IBM Keyboard Substitute Version

The microcontroller scans the keyboard 189 which is made up of the normal QWERTY keys and the function keys as follows:

Bits 0-5 of part 1 are used to set up the address of (1 of 24) a keyboard row which is decoded by the LS154 and LS138 decoder devices. The addressed row is then strobed by the matrix strobe line which is controlled by bit 5 of part 3 of the microcontroller. Simultaneously with the row addressing the 1 of 4 column addressing is carried out by the 4051 multiplexer controlled by bits 6, 7 and 8 of part 1. The row and column addressing timed with the matrix strobe line will result via a 2 amplifier (LM347) detector and 7404 invertor in an active low pulse on the key depress line.

The reason for the dual amplifier detector is that the keyboard matrix is capacitance coupled and subsequently requires supporting technology.

Item 195 is a row of pullup resistors. Item 194 provides the function of rollover enable.

(ii) & (iii) Interface Unit and RS232 Peripheral Versions

In both these versions the functions keys are normally

open switches and therefore require no elaborate detection. The keyboard is scanned on the five rows by bits 0-4 of part 1 and on the columns on bits 2 and 3 of part 3.

5. Communications Interface

(i) IBM Keyboard Substitute Version

The keyboard interfaces via 5 lines to the personal computer (192) two power supply (+5V,OV) from which power for the keyboard circiutry is derived and three control lines. These 3 lines are as follows:

Pin 3- hardware active low reset is controlled by the IBM-PC and resets the keyboard microcontroller via a logic invertor to pin 9.

Pin 2- bi-directional serial communications line which is connected to bits 0, 1 of part 3. These pins on the microcontroller are special function pins and are implemented internally by the microcontroller as receive data and transmit data for serial communications.

Pin 1- bi-directional control line which is used to transmit the serial communications clock in both directions and also to implement a single bit control logic. It is connected to bit 4, part 3.

(ii) RS232 - Peripheral Version

In this version all external communications are carried out via EIA standard RS232C serial port (196). Essentially the microcontroller implements the transmit data and receive data bits 1 and 0 respectively of part 3

for the full duplex serial data path. Bits 4, 5 part 3 and 5, 6 of part 1 are used to control the hardware protocol lines DTR, DSR, CTS and RTS in accordance with the EIA standard. All lines connecting to the EIA RS232C standard have to conform to specification of voltage and logic levels. For this reason the 5V positive true logic on the microcontroller has to be buffered by the 1488 and 1489 line buffer devices which operate as $\pm$ 12V negative true logic. Item 198 is a reset circuit and a 5V power supply (197) is used as no power line is provided on the RS232 line.

(iii) Interface Unit Version

This version has two communication parts. The first are interfaces to the IBM-PC and is identical to the one described in (i) above. The second (191) interfaces to an IBM-PC standard keyboard. This part consists of 5 lines. The +5V, OV (193) and reset lines are connected directly to the IBM-PC part for continuity. Pins 1 and 2, the keyboard clock and keyboard data lines respectively interface to the microcontroller via bits 4 and 5 of part 3. Unlike the bits 0 and 1 of the microcontroller which perform as hardware serial lines these pins will perform the necessary protocol under software control.

Means for Interfacing Function Keys

The major features of the means for interfacing one or more tactile function keys where each key comprises a single or multicharacter non-specific display as implemented are:

1) Method to display characters on the function keys.

2) Method to change current displays on the function keys.

3) Method to define and change character sets available to be displayed on the function keys.

4) Method to "ghost" host a set of conventional actions upon the depression of a function key via the host processor as though they were entered in the normal way, without interfering with the main level program in operation on the host computer.

5) Method of signalling to the host processor that a function key has been depressed, and executing upon the host processor a procedure to handle the function key depression.

6) Method to co-ordinate keystrokes from both the computer keys and the function keys, and send them to the host processor as though they all originated at the computer keyboard.

7) Method of monitoring all conventional keystrokes executed, and searching for a character string matching that assigned to a particular Function key, and then changing the function definitions on each function key, and the keystrokes they "ghost" as if that particular function key was depressed.

8) Method for handling all "ghosting" and "function key equivalence checking" within the keyboard or keypad during the execution of a main level program so that once the function key definitions and keystrokes to be "ghosted" for all sets of displays and function levels assigned a program

have been downloaded to the keyboard before execution of the main level program, the host processor will receive no distinction as to whether the keystrokes were executed manually or "ghosted" by a function key, and the function key descriptions will be updated in a conventional manner.

9) Means for interfacing the function keys to dumb electronic equipment.

10) Computer system designed around function keys.

11) Method of storing data for keystrokes to be "ghosted".

In order to display characters on the function keys, a RAM (may be substituted for a ROM in the case of the unit to interface with "dumb" electronic equipment) is used to store codes for the characters that are currently displayed on each key surface. A ROM, EE ROM or a ROM with an additional selectable RAM is used to store the information necessary to define each character in the character set currently being used for its display on the display format being used eg: which dots should be on or off to display an "A" on a dot matrix display. Using a timing circuit the process constantly loops, taking characters from the RAM, encoding them using the data in the ROM (or EE ROM or ROM with an additional selectable RAM), serialising the data, and sending it to the displays with addresses in the format they require. The displays may be strobed ie: taking the top row of all characters being displayed at a particular time, and then the next row etc., or sent to the displays in whatever manner is required.

In order to change the function definitions currently

being displayed it is necessary only to the display cycle and dump the required data to the RAM, and then resume normal operation.

The character set available to be displayed on the function keys can be redefined providing this information is stored in a EE ROM or a ROM with an additonal selectable RAM is being used. If an EE ROM is being used, the display cycle must be stopped, a signal sent to the EE ROM to refresh it and allow the new data to enter, and then the new data must be downloaded to it. The display cycle can then resume. If a ROM with an additional selectable RAM is beng used, the display cycle must be stopped, the new information downloaded into the RAM, and the RAM selected as the default source. The display cycle can then resume.

In order to "ghost" a set of conventional keystrokes upon the depression of a function key via the host processor, the host computer must be designed around interrupt logic. The method works by sending an identifyable interrupt, different from that generated by the computer keyboard. Prior to this the processor has been instructed that upon receipt of such an interrupt it should execute a routine which will generate an interrupt by software identical to those sent from the computer keyboard when a conventional key is depressed and make each item of data for the keystrokes to be "ghosted", available in the same manner as they would come from the computer keyboard.

If the host processor is to be used to handle the

"ghosting" process, once the identifyable interrupt signal has been received from the function keys, the processor must run the required procedure. Generally upon receipt of an interrupt, the process will access a special area of memory that contains the starting address of the procedure. By allocating an area of memory at initialisation specifically for the procedure, and storing the starting address for the area at the required place and storing the procedure in the area allocated, the processor will jump to and execute the correct procedure upon receipt of the interrupt.

If one of the function keys are depressed, an interrupt signal is sent to the host processor and a code is made available to it that indicates which function key was depressed. Upon receipt of the signal the processor executes a procedure that accesses the code. The processor has access to data stored in the memory of the host computer that contains information for a number of sets of function definitions to be displayed on each key, and keystrokes to be "ghosted" when each function key is depressed. The data also contains the number of the set to be jumped to when each key is depressed. The processor uses this information and the code for the key depressed, and if a new set is to be jumped to it such information to the memory on the keyboad which will change the function definition displayed and the keystrokes "ghosted".

In order to co-ordinate both keystrokes the computer keys and the character to be ghosted upon depression of

function keys, a microprocessor is used to receive the signals from each, process them in the manner required (ie. converting the signal sent from a function key to the required ghost keystrokes and performing "function key equivalence checking") and then send them in an identical manner along the keypad cable to the host computer or terminal. In the case of the unit that intercepts the keyboard cable, the signals from the computer keys will be obtained from the keyboard cable which is redirected to plug into the peripheral, and all signals to the host device are sent along the keypad cable. The data will be buffered and work via interrupts or handshaking conventions so that no data will be lost or contaminated unless the buffer overflows.

The logic used to check if the set of keystrokes that are defined to be input by any function key is input manually works in the following manner:

1) An area of memory is defined and used as a "circular" buffer to store the characters sent from the computer keyboard sequentially. The buffer is described as circular as if the end of the buffer is encountered, the process continues, from the start of the buffer, and if the start of the buffer is encountered moving backward, the process continues from the end of the buffer.

2) A pointer is used to indicate the current position within the buffer at any time.

3) When a conventional key is depressed the following steps are followed:

i) The character is sent through to the host processor.

ii) The pointer is advanced one character.

iii) The characters in the buffer are compared with each of the sets for the keys.

For each key in turn:

1) The latest keystroke entered is compared with the last character in the set of characters for that key.

2) If the characters match, the character immediately before that last tested in the buffer is compared with that immediately before that tested in the set for the key.

3) This process is repeated until all of the characters in the set for the key have been matched (ie. the end of the set is reached) or a pair of characters that do not match are encountered.

4) If all of the characters for the set for the particular key are matched and the host processor is being used to process signals from the function keys directly then an interrupt signal is sent to the host processor, identical to that sent if one of the function keys were depressed itself, and a code to indicate which function key was depressed is similarly made available to the host processor. If the microprocessor or the keyboard/keypad is being used to process the signals from the function keys so that only convention keystrokes are echoed to the host device, then it will execute the same actions as it would if the particular function key were depressed. The buffer is then cleared, and the process starts from step (i).

In order that no requirement is made on the host processor to be structured around interrupt logic, and so as to not use memory space on the host computer or divert the host processor to excede the required interrupt procedure upon the depression of a function key, additional memory (RAM) can be added to the keyboard to allow the data for all sets of function definitions to be downloaded to the keyboard before executing the main level program. The processor on the keyboard has access to all of the necessary information and can execute the "ghosting" and "function key equivalence checking" itself. The processor will pass all keystrokes, whether input from the computer keyboard, or "ghosted" from the function keys to the host computer, as if they all came from the computer keyboard. Such a keyboard will be able to be used with any computer that allows transfer of data from the computer back to the keyboard after modifying the plug on the keyboard and the key code signalling system used to that required by the computer so that one unit can be used for all computers with the same plug that allows data transfer back to the keyboard, the firmwear program that operates the processing unit the keyboard can be designed to download the necessary software for its operation from the host computer upon initialisation. Only the software would need to be changed between units.

A keypad may operate in the same way if the keyboard plug from the computer keyboard is diverted to plug into the keypad. The keypad would then have access to

information from both the computer keyboard and the function keys and would send its key codes back to the host computer through its own cable, which can plug into the socket for the standard keyboard plug.

A keypad may also operate in the same manner using any bi-directional communication system such as an RS232, if a routine is incorporated to work with the operating system of the computer to make it echo every keystroke received from the computer keyboard through the communication line being used so that it is available to the processor on the keypad and allow the processor to accept characters sent to the RS232 port on the computer Terminal as input in the same manner as keystrokes from the computer keyboard are accepted. The key signals from the keyboard may be allowed to pass directly to the processor with an echo being sent down the RS232 or, the keystrokes may be redirected so as not to be recognised by the host process as input, but rather sent straight to the processor on the keypad and sent back in the same manner as those characters ghosted from the function keys. Almost every computer or computer terminal has a bi-directional port (eg. bi-directional keyboard plug, RS232, RS132) through whch such a system could gain access. Thus a keyboard or keypad using one of the methods above could interface to almost every computer or computer terminal available. It is not required that such a unit execute the functions of "function key equivalence checking" and it may be available as an option. If "function key equivalence checking" is not used

it would not be necessary to echo every keystroke from the computer keyboard down the RS232 cable to the keypad.

A keypad or keyboard can interface with "dumb electronic equipment" not designed for use with it provided it has means for simulating the manner in which the functions represented by each key would be executed manually. This would be especially useful in testing equipment. Many items of "dumb electronic equipment" may be redesigned around use of the function keys eg. cash registers. A microprocessor would control the function definitions and functions executed by each key, in the same manner as for the system above, however the data for the function definitions and steps to be executed would be stored in a ROM, or EE ROM, or in a RAM that is initialised when the device is. The depression of each key may result in whatever type of signal that is necessary to execute that function eg. millivolt, digital etc. The processor may also be used for other tasks eg. cumulative totals on a cash register.

Programs that are specifically written for use with the function keys will not need the "ghost writing" or "function key equivalence checking facilities". These programs will immediately access the fact that a function key has been depressed, and identify it. The program will be designed to take action on this depression. Such programs will directly control the function definitions on the displays by sending the required data to the RAM on the keyboard/keypad. Thus the keyboard/keypad will operate in

two selectable modes ie. under the direct control of the main level program, or under the control of the "ghosting"/"function key equivalence checking system".

A computer system may be designed around the function keys of the present invention or one may be available with sufficient flexibility to have the function keys simply input a code in the same manner as any conventional key and have the host processor recognise it as one from the function key and perform the task of checking which characters should be input to the program running, and then reconfiguring the display on the function key.

So as to make the optimum use of the memory code available, the information for the function definition and the keystrokes to be "ghosted" for each key are stored dynamically. Rather than allowing a set of number of characters to be ghosted for each key in each set, a system of pointers is used as shown in Fig. 14. The first element of the data 400 series contains the number of sets, then a pointer to the start of each set. A group of data for each set then follows, the first item being pointed to the start of the information for each key, and then a package of information comprising the function definition to be displayed on the key, the set to jump to when the key is depressed, the number of characters or actions to be "ghosted", and the characters or actions to be "ghosted" follows.

THE CLAIMS:

1. A function key comprising a single or multicharacter non-specific display, means for activating said display, and means for interfacing said function key with an electronic system.

2. A function key comprising a single or multicharacter non-specific display located thereon, means for activating said display, means for interfacing said function key with an electronic system, said function key being programmable so as to alter the function of the key to that required and said display being programmable to alter the display as required with respect to the function output of said key.

3. A peripheral keypad comprising a plurality of function keys according to claim 1 or 2.

4. A keyboard that includes a keyset comprising a plurality of function keys according to claim 1 or 2.

5. A keypad or keyset comprised of one or more tactile function keys, each key comprising a single or multicharacter non-specific display, means for activating said display, and means for interfacing said function key with an electric system so that while running a main level program, each function key can be used to "ghost" a set of signals to the host electric system as if they were generated by means conventional to the system, and both the function definitions displayed on each function key, and the characters "ghosted" when each function key is depressed can be changed for each function key without affecting the execution of the main level program.

6. A keypad linked to, or keyset incorporated in, a computer keyboard, said keypad or keyset comprising one or more tactile function keys, each key comprising a single or multicharacter non-specific display and having means to monitor the conventional keystrokes being input from the computer's keyboard, and on entry of the set of characters defined as input upon depression of any one of the function keys, the function definitions displayed on each function key, and the keystrokes to be "ghosted" upon the subsequent depression of a function key may be changed in an identical manner as if that particular function key were depressed instead.

7. A keyset comprising one or more function keys according to claim 1 or 2 wherein said function keys are tactile, wherein said interfacing means interfaces said function keys with a "dumb" electronic system, and including program means to control the operation of said "dumb" electronic system.

8. A function key according to claim 1 or 2 having a display screen associated therewith wherein output of said key can be interfaced to any computer with available data path or paths in both directions.

9. A function key according to claim 8 wherein said display screen is located on said function key.

10. A peripheral keypad having a plurality of function keys according to claim 8 or 9.

11. A function key according to any one of claims 1, 2, 8 or 9 wherein the display is a Liquid Crystal Display.

12. A function key according to claim 11, wherein the display is driven by chip on glass technology.

13. A function key according to any one of claim 1, 2, 8 or 9 wherein the display is a Light Emitting Diode.

14. Tactile key comprising:

a moveable member operated by the user;

a display mounted on said moveable member and visible to said user;

a switch mechanism adapted to be operated by movement of said moveable member; and

connector means adapted to connect said display to a source of control signals, which connector also provides connection therebetween, during movement of said moveable member.

15. A function key according to claim 11, wherein display is driven from a printed circuit board mounted below said display and connections being made by zebra stripes.

16. A function key according to any one of claims 9, 11, 12 or 13 or 15, further comprising the features of claim 14.

17. A keyboard or keypad comprising a plurality of keys according to any one of claims 14 to 16.

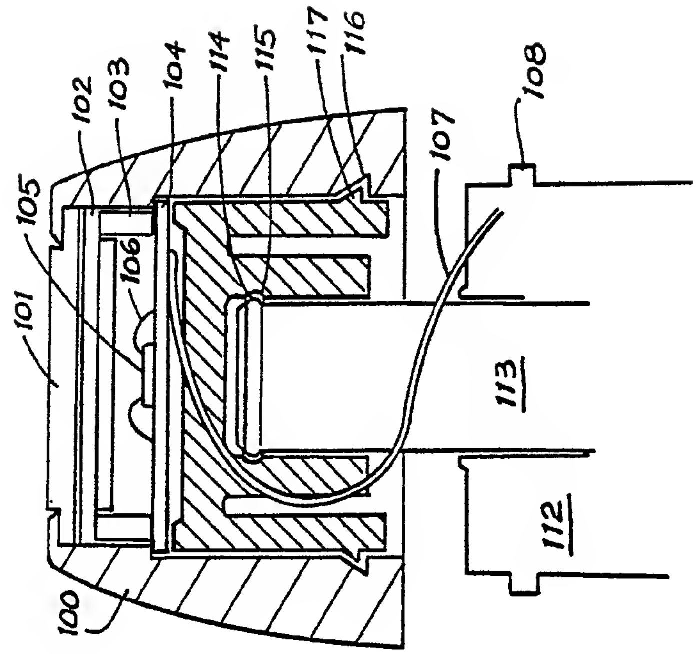

100
101
102
103
104
105
106
107
108
112
113
114
115
116
117

FIG.2

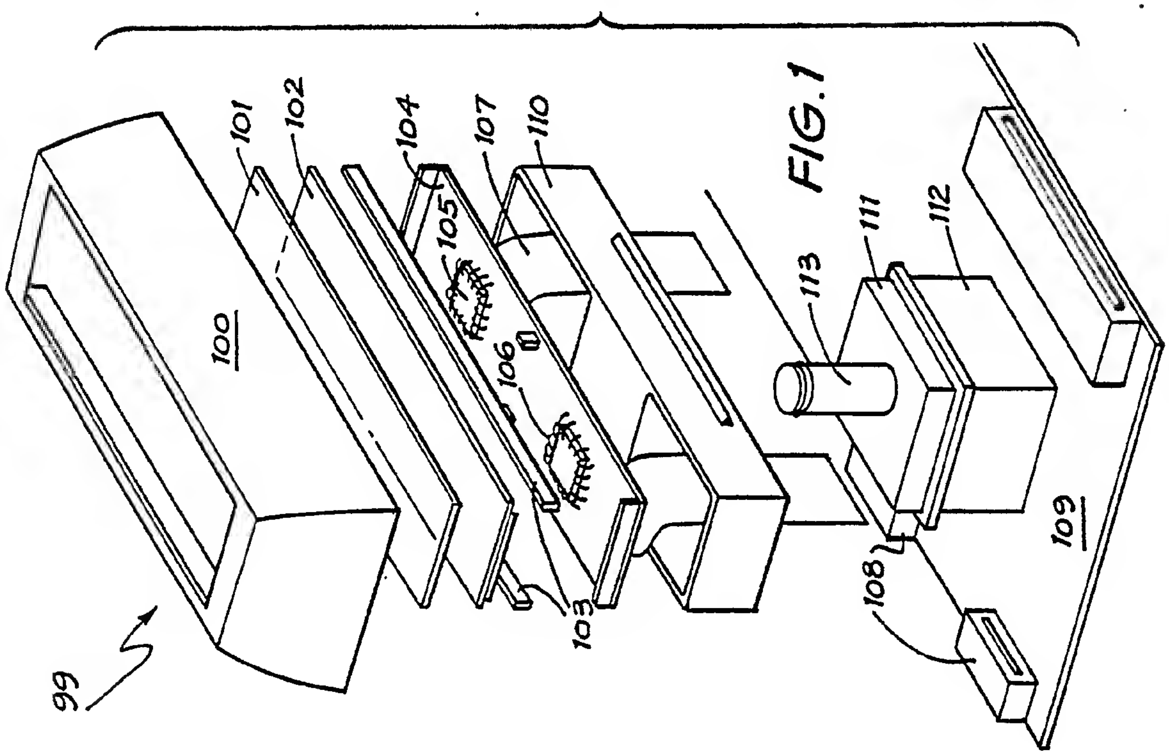

99
100
101
102
103
104
105
106
107
108
109
110
111
112
113

FIG.1

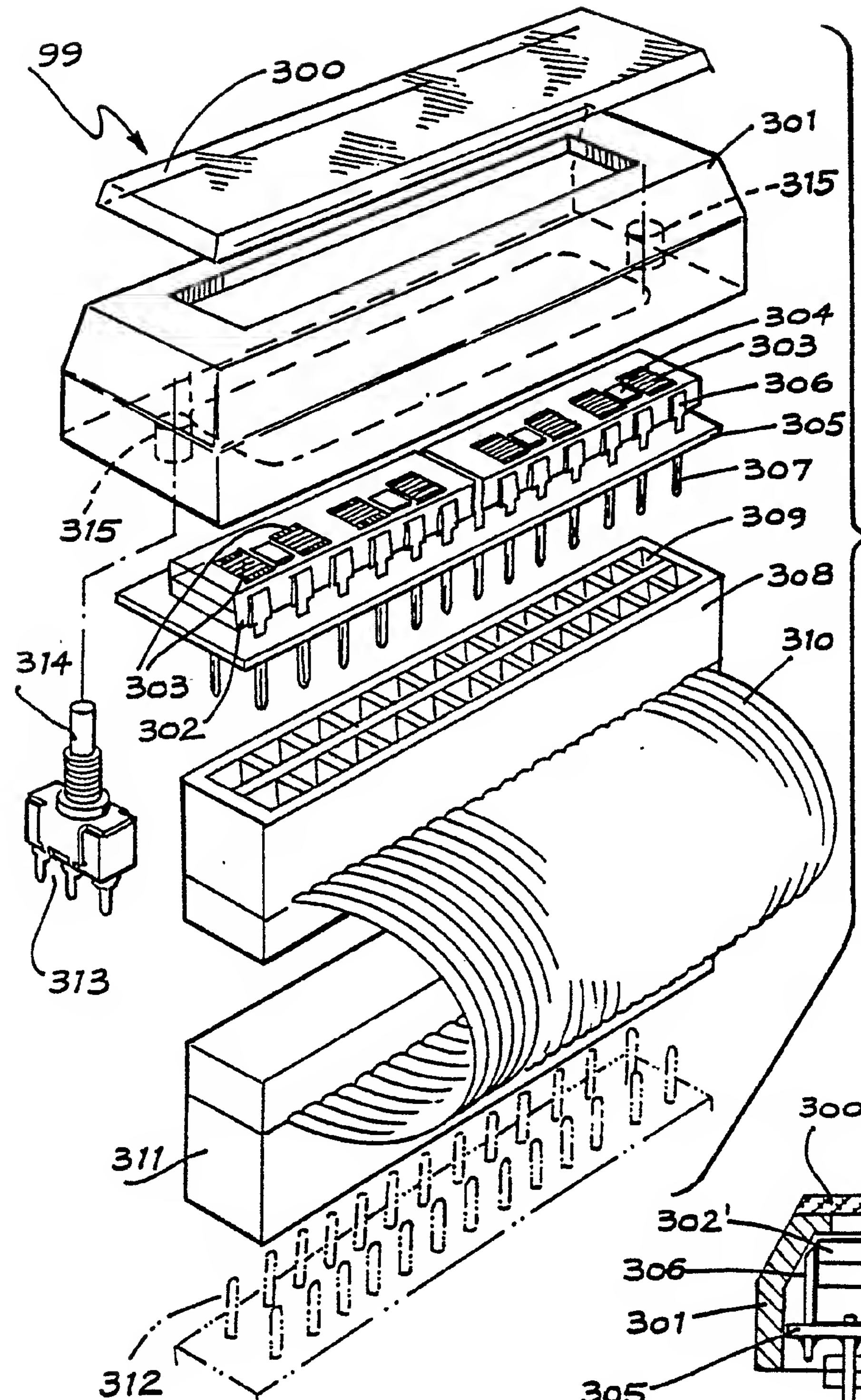

99
300
301
315
304
303
306
305
307
315
309
308
310
314
303
302
313
311
312

FIG.3

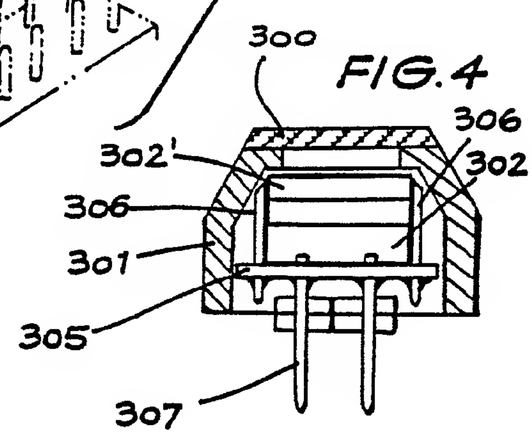

300
306
302'
302
306
301
305
307

FIG.4

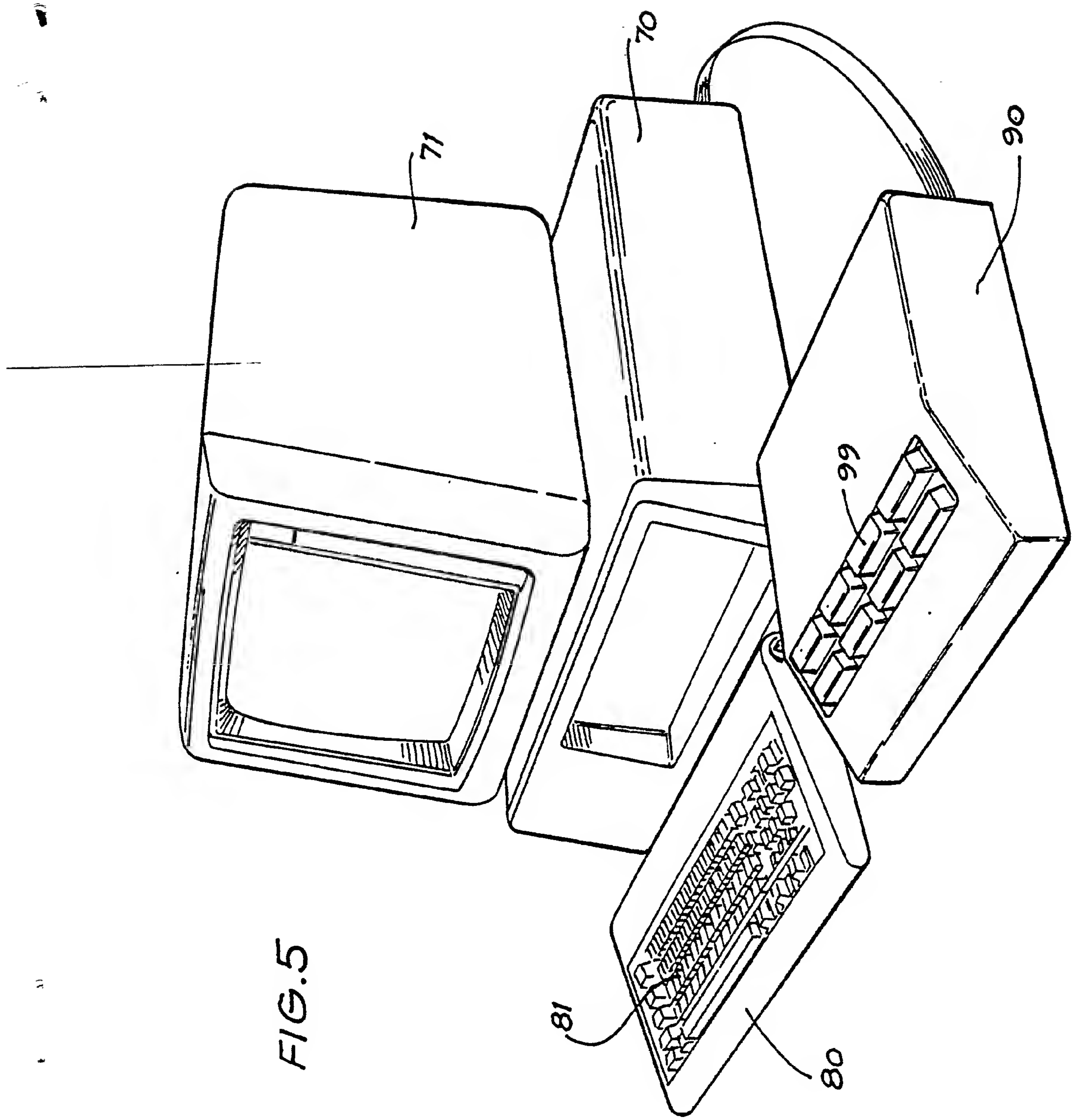


FIG.5

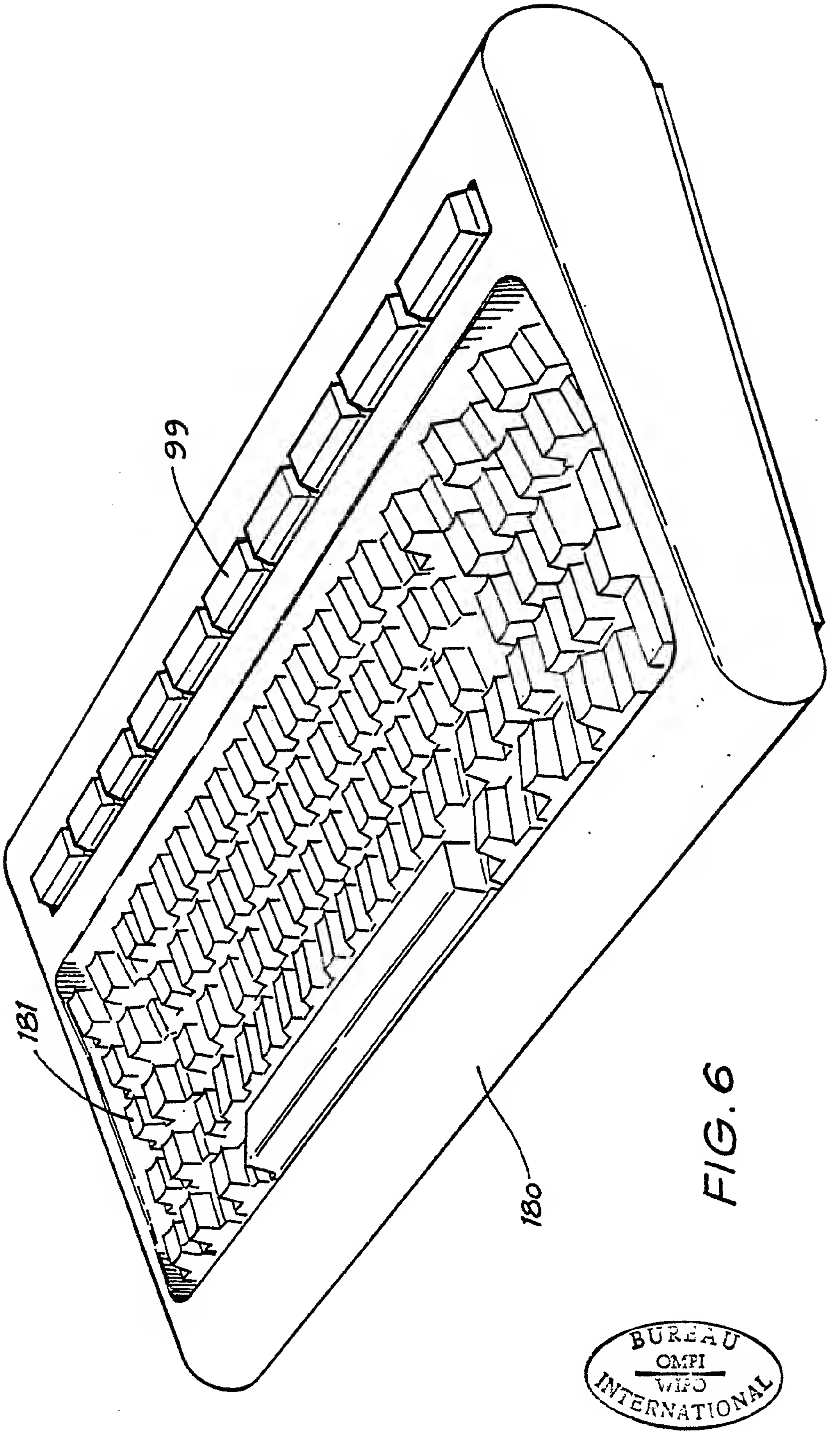


FIG. 6

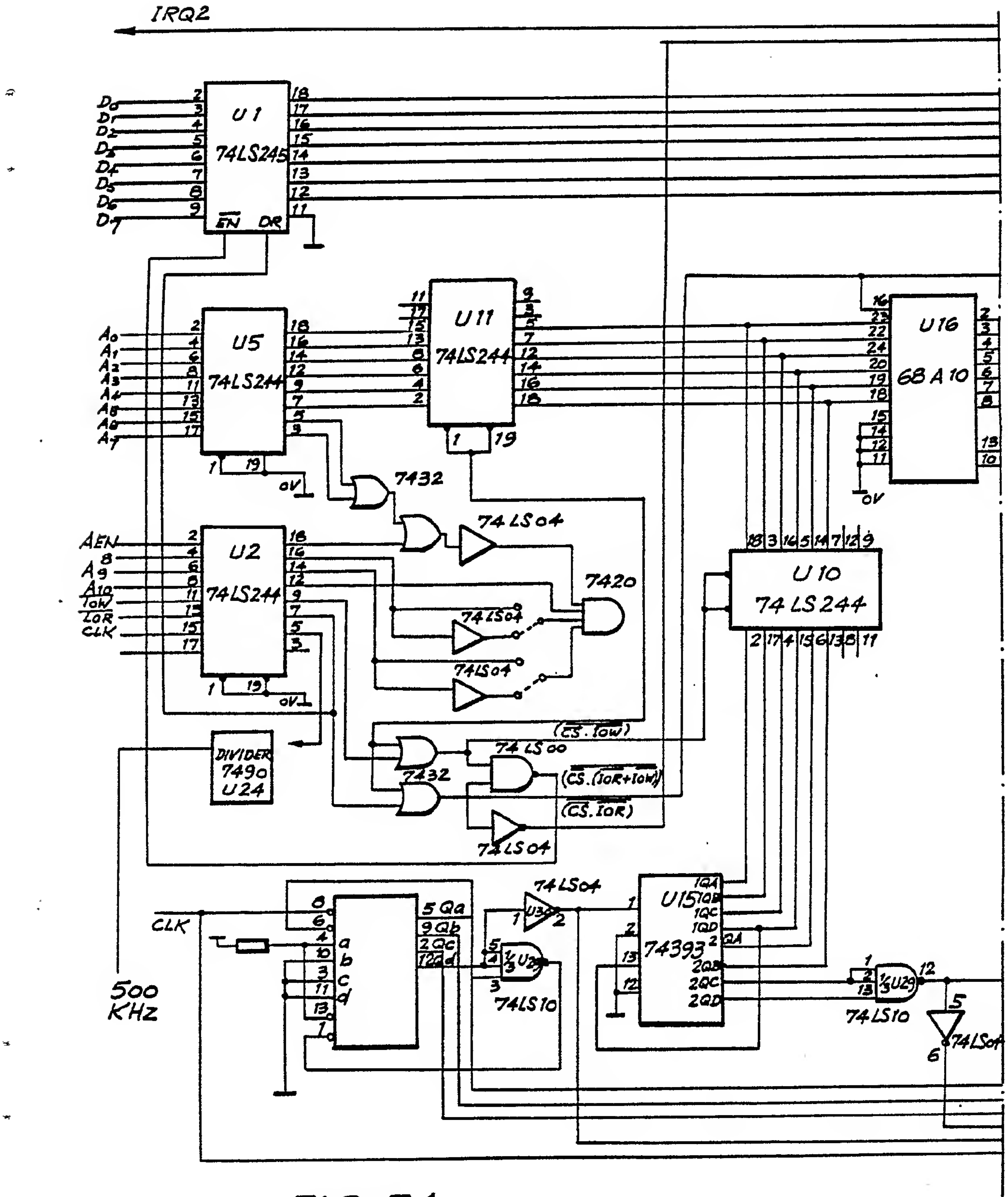

IRQ2
U1
74LS245
EN
DR
U5
74LS244
U11
74LS244
U16
68A10
U2
74LS244
AEN
CLK
0V
7432
74LS04
7420
U10
74 LS 244
74 LS 00
(CS.IOW)
(CS.(IOR+IOW))
(CS.IOR)
DIVIDER
7490
U24
500 KHz
U15
74393
74LS10
74LS04

FIG. 7A

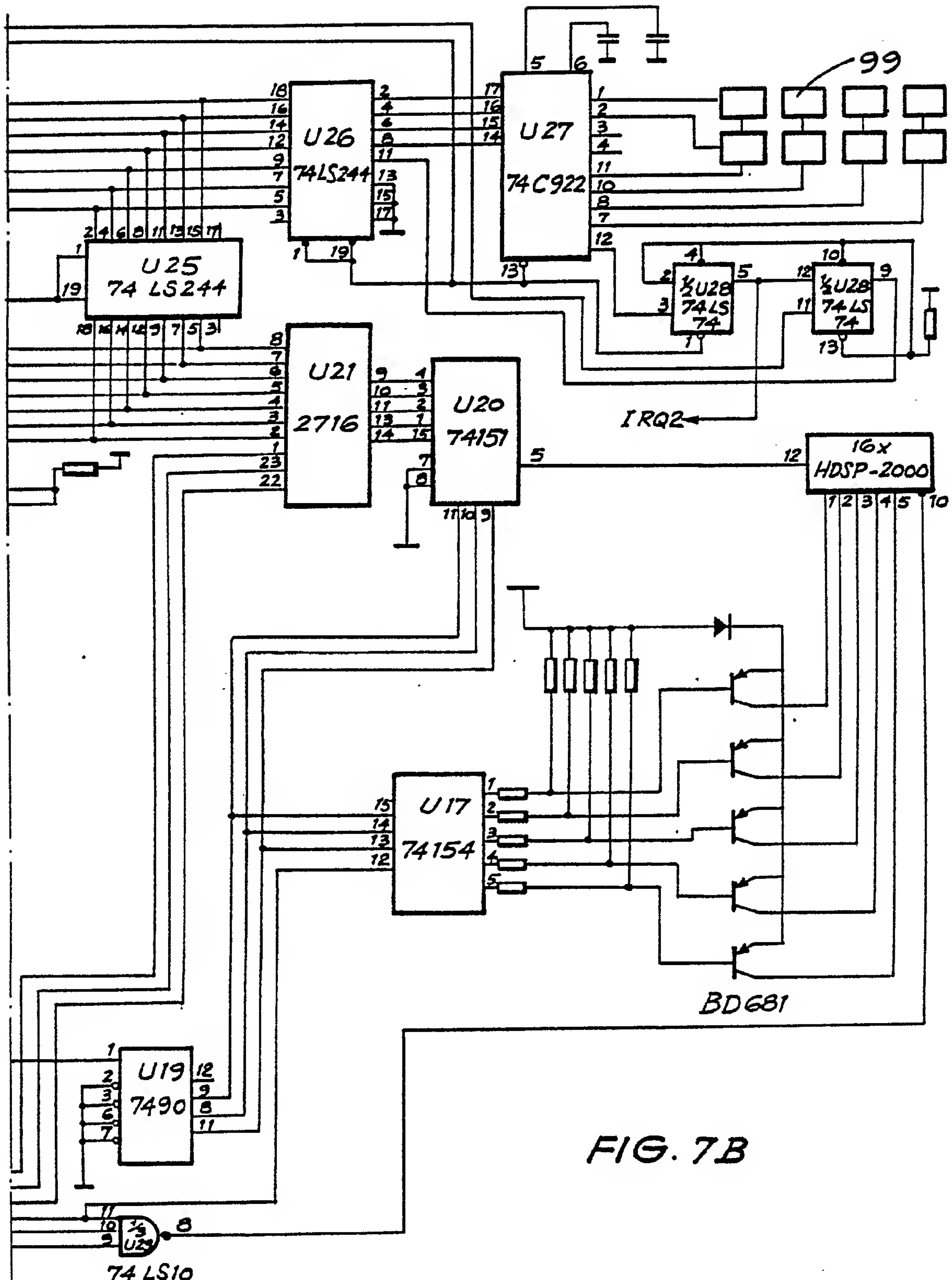

99
U26
74LS244
U27
74C922
U25
74 LS244
U21
2716
U20
74151
1/2 U28
74 LS
74
IRQ2
16x
HDSP-2000
U17
74154
BD681
U19
7490
1/3 U29
74 LS10

FIG. 7B

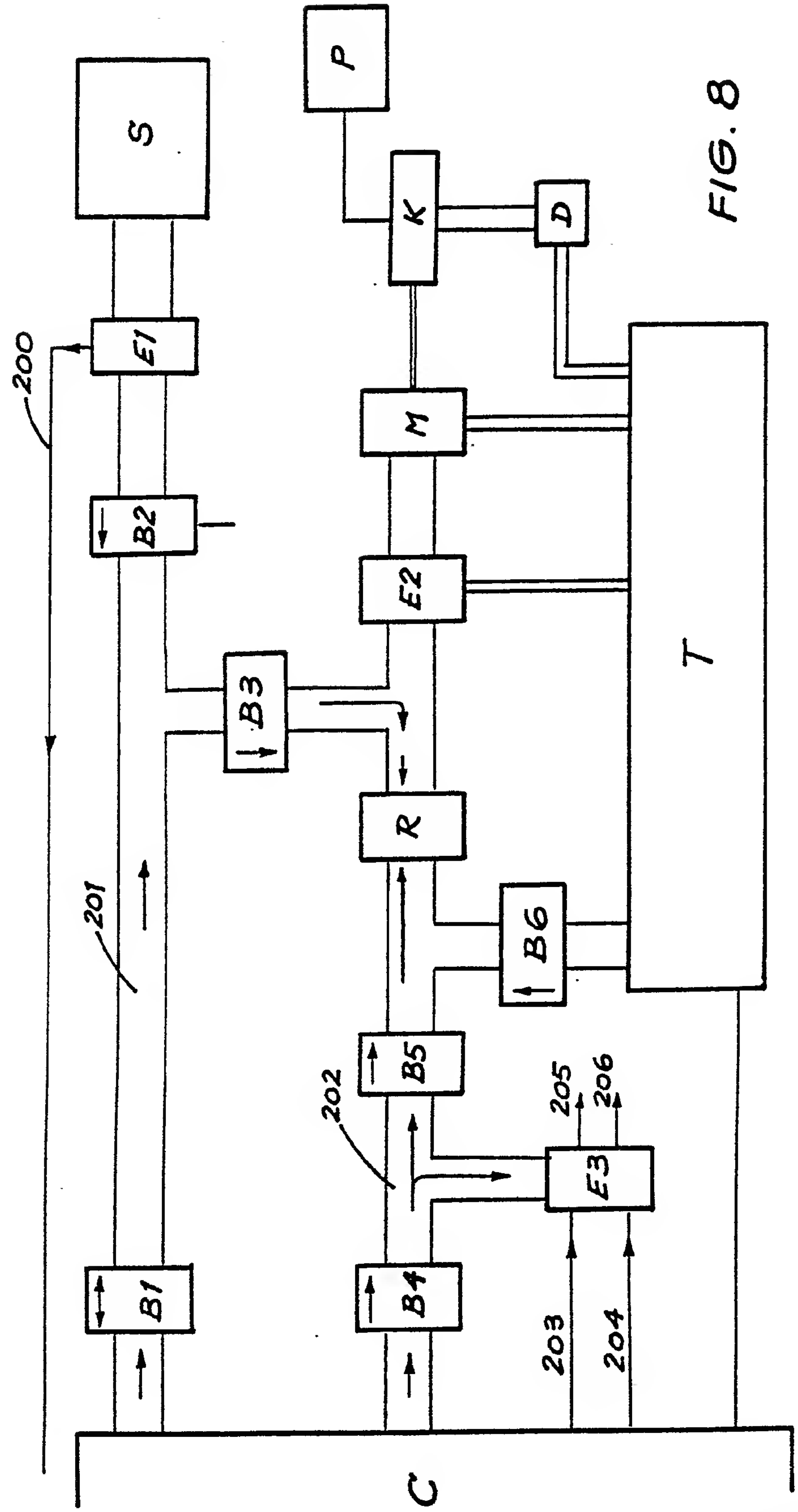


FIG. 8

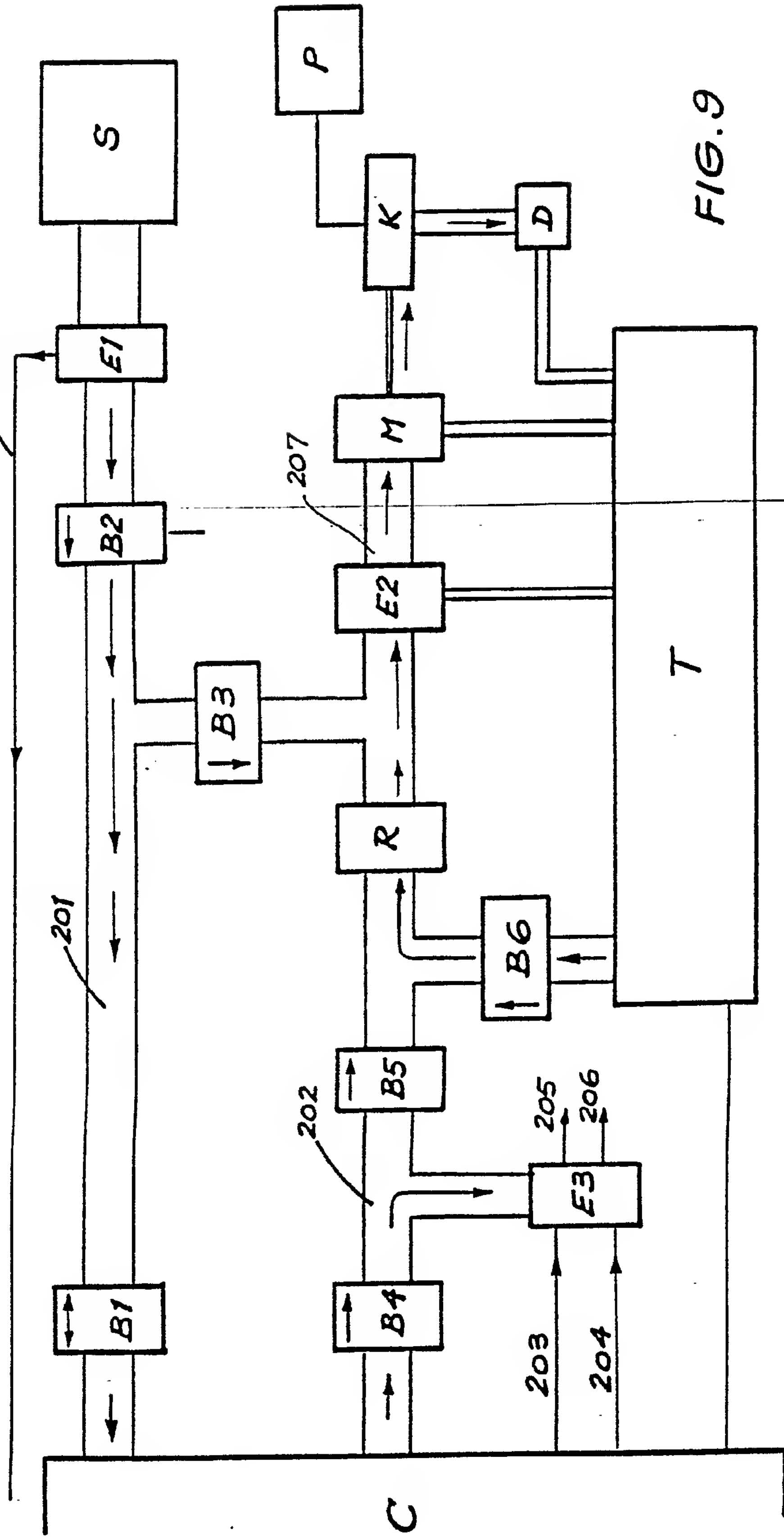

S
P
K
D
E1
B2
B3
E2
M
T
R
B6
B5
E3
B1
B4
C
200
201
202
203
204
205
206
207

FIG.9

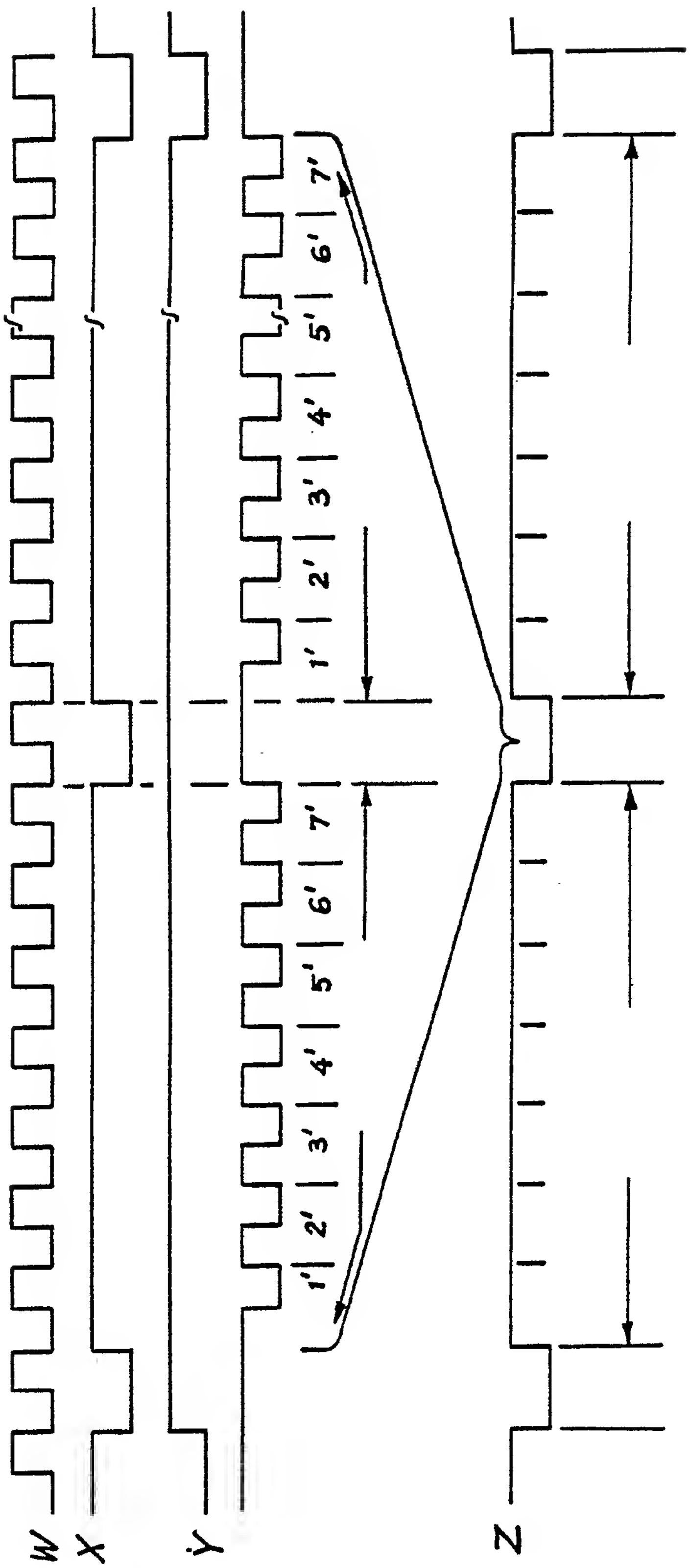


FIG. 10

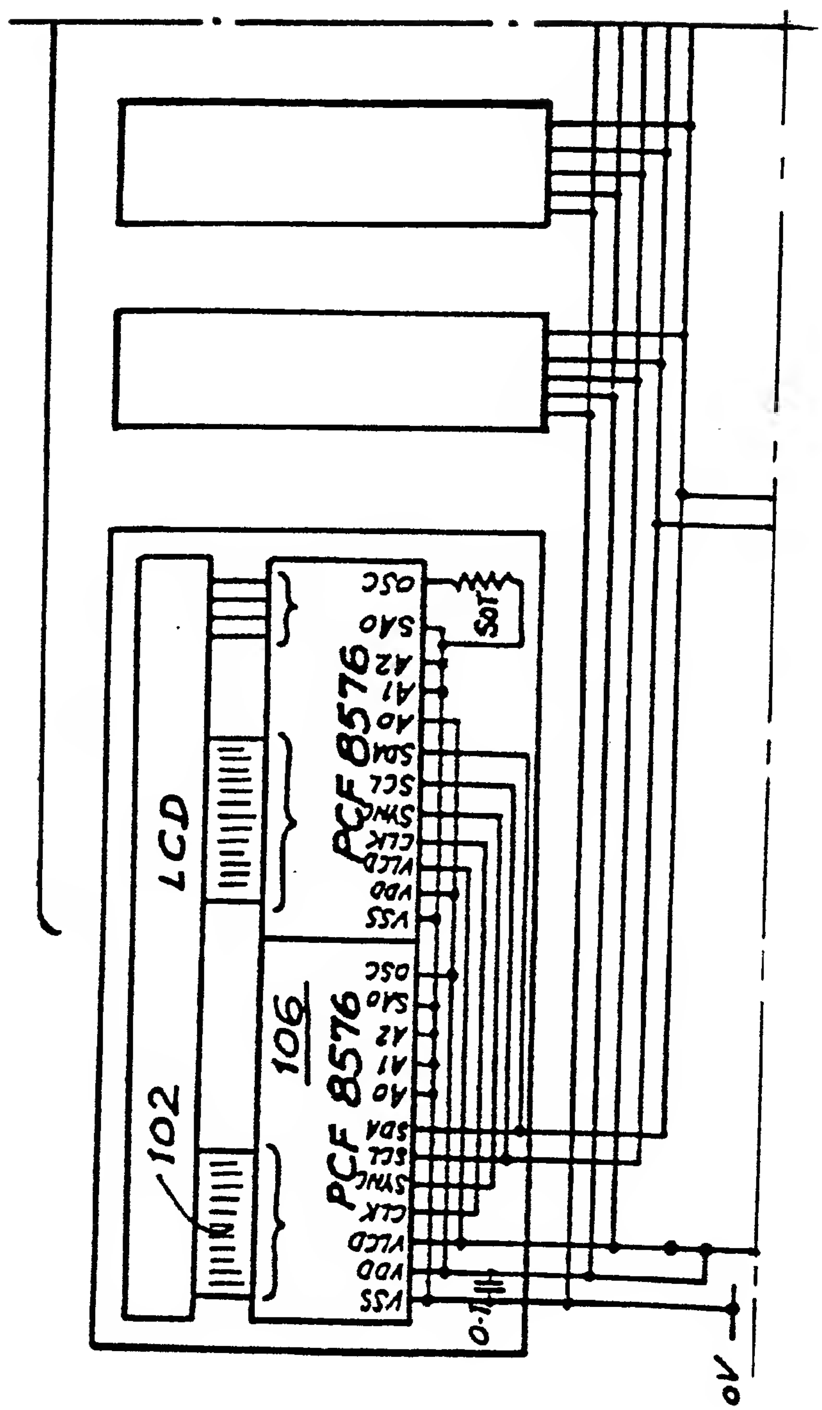

LCD
102
106
PCF 8576
PCF 8576
OSC
SAO
A2
A1
A0
SDA
SCL
SYNC
CLK
VLCD
VDD
VSS
0V

FIG. 11a'

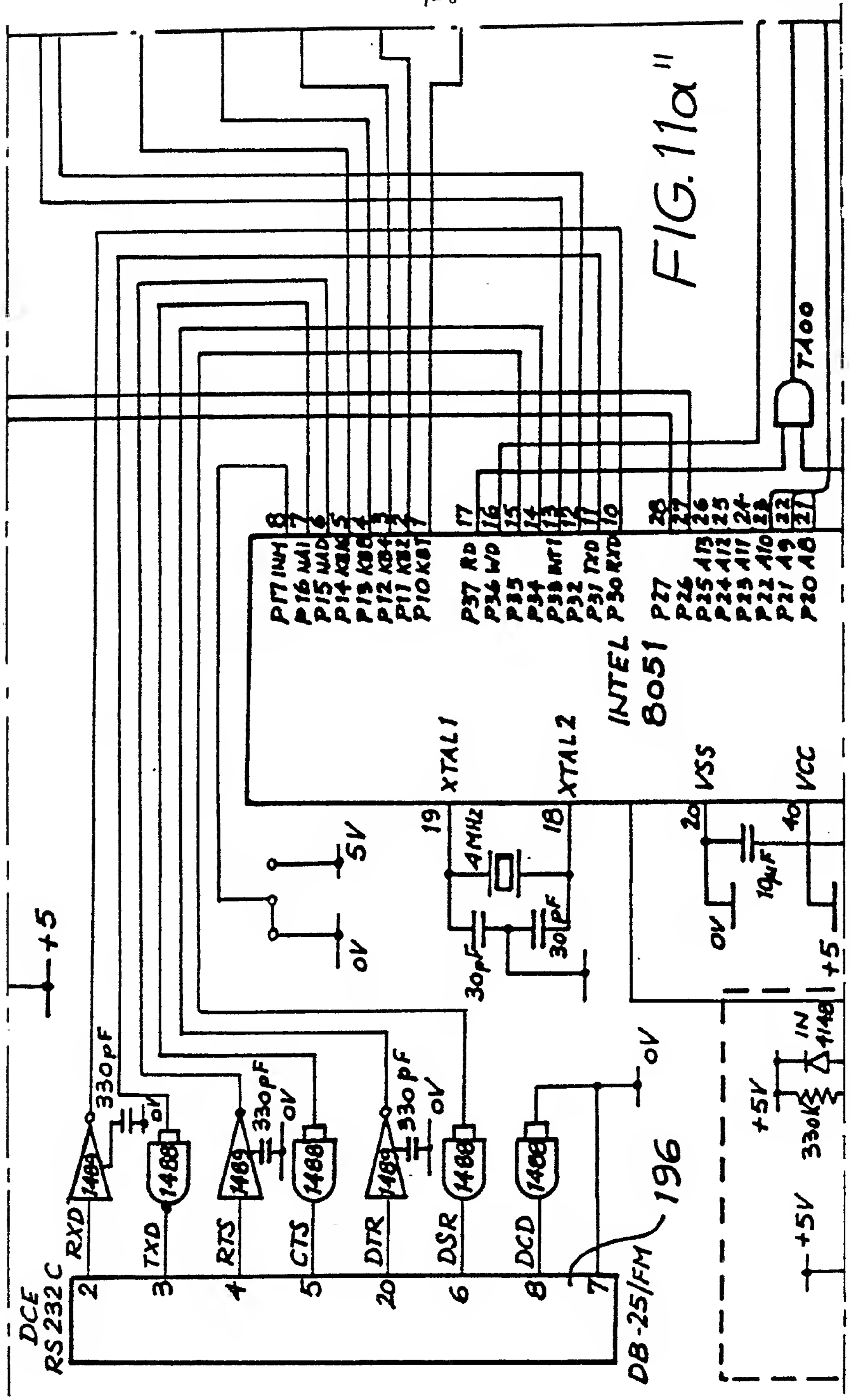

INTEL
8051
XTAL1
XTAL2
VSS
VCC
4MHz
30pF
10μF
1N
4148
330K
+5V
+5
0V
5V
TA00
1489
1488
330pF
RXD
TXD
RTS
CTS
DTR
DSR
DCD
DCE
RS232C
DB-25/FM
196

FIG. 11a"

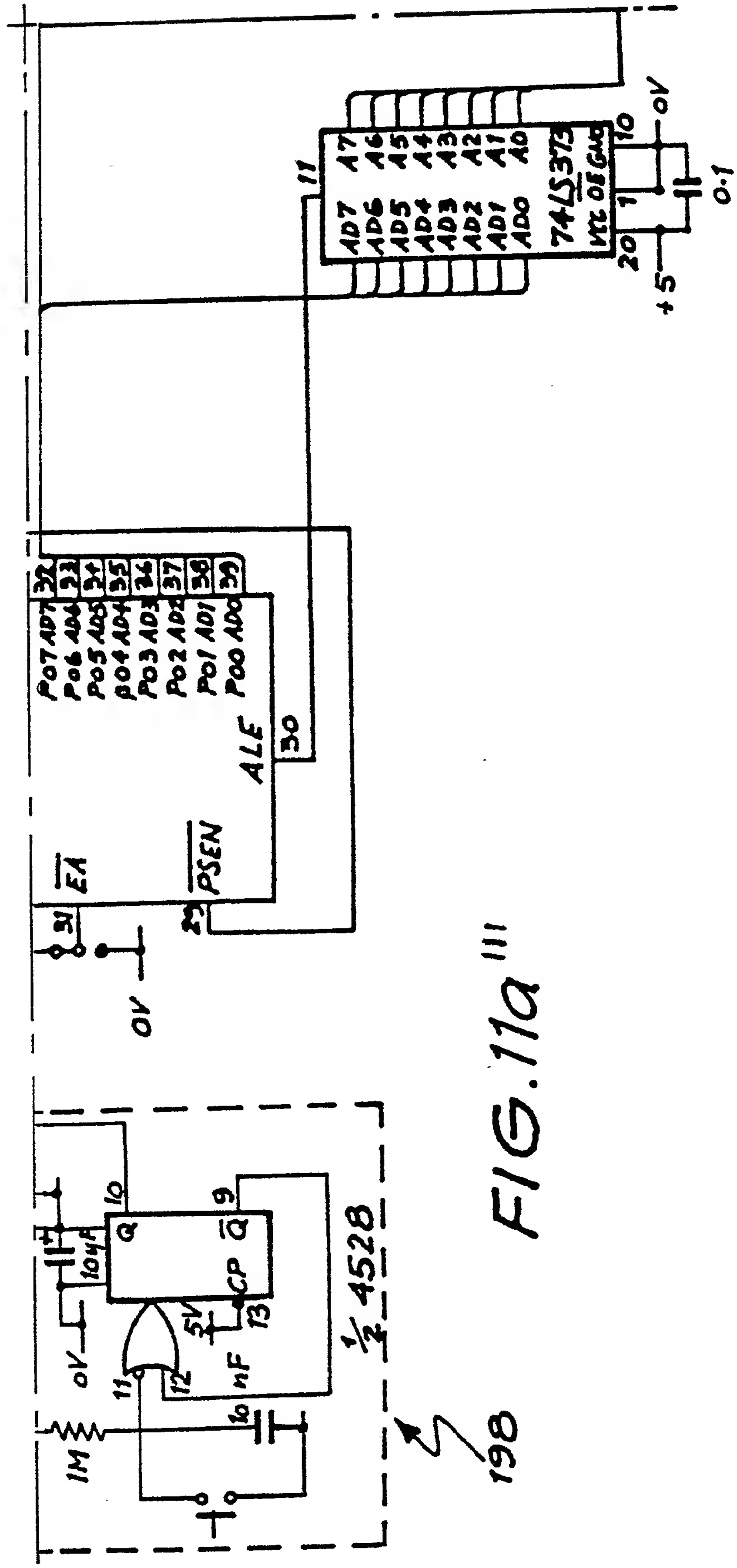


FIG.11a'''

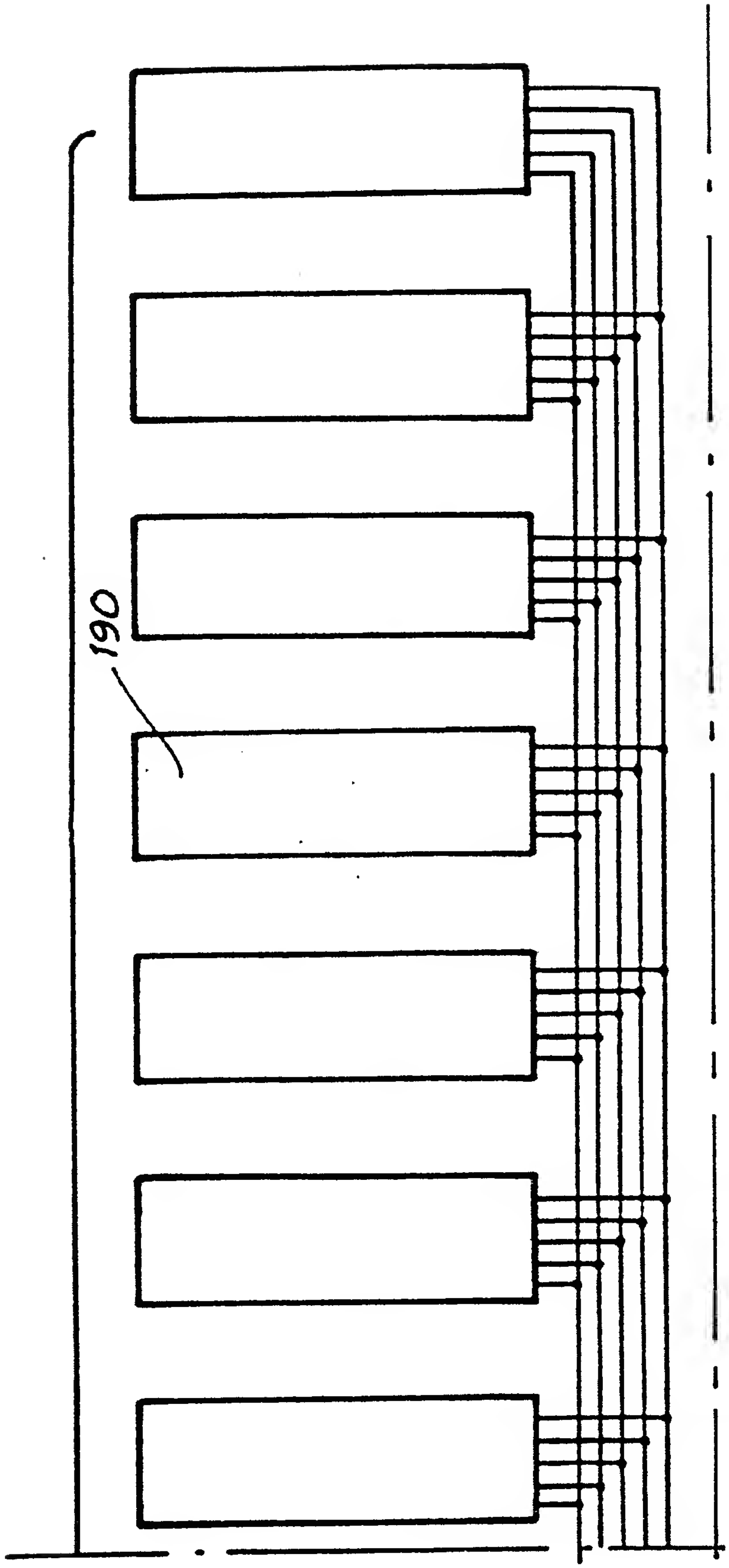


FIG.11b'

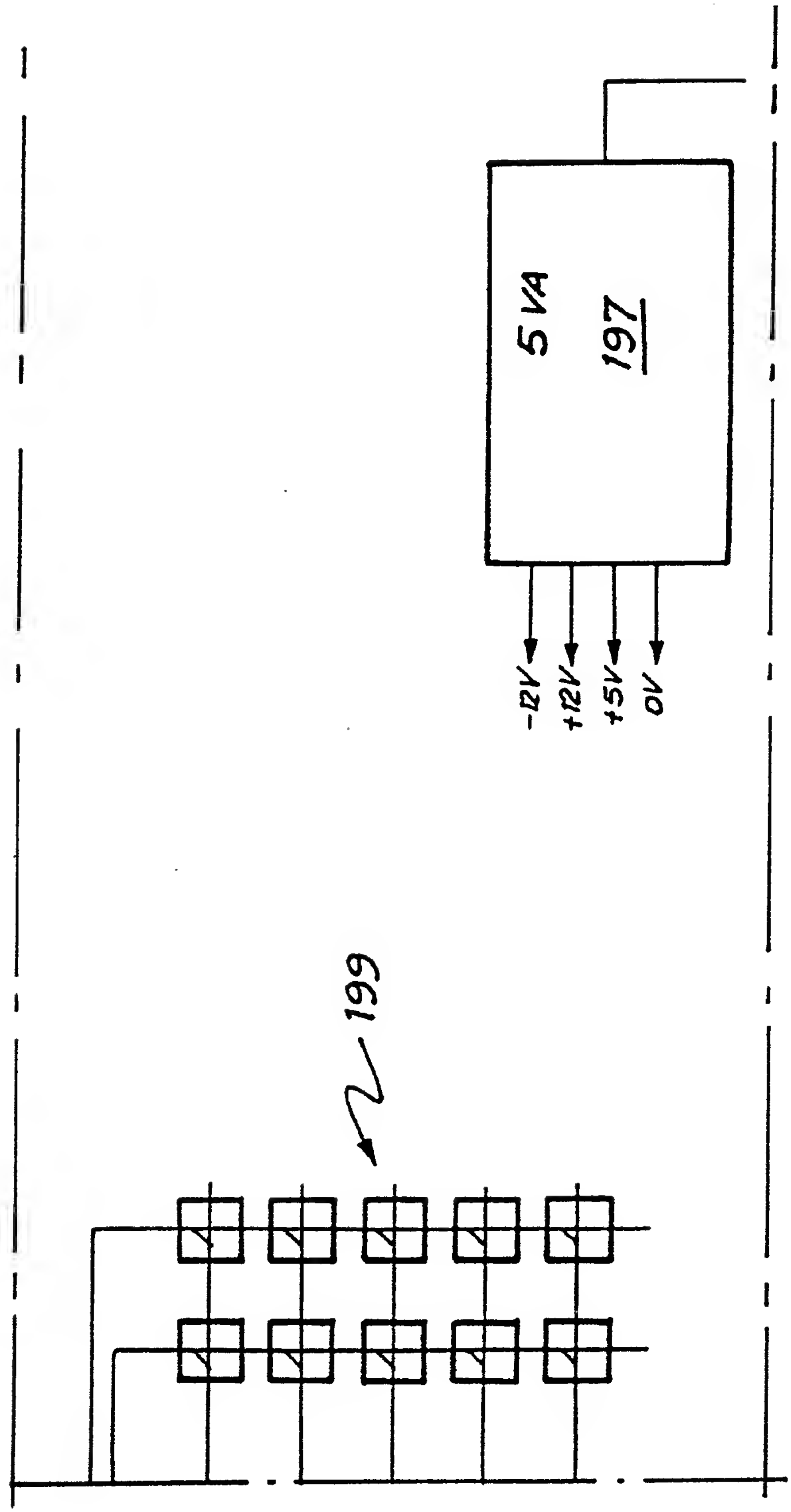


FIG.11b''

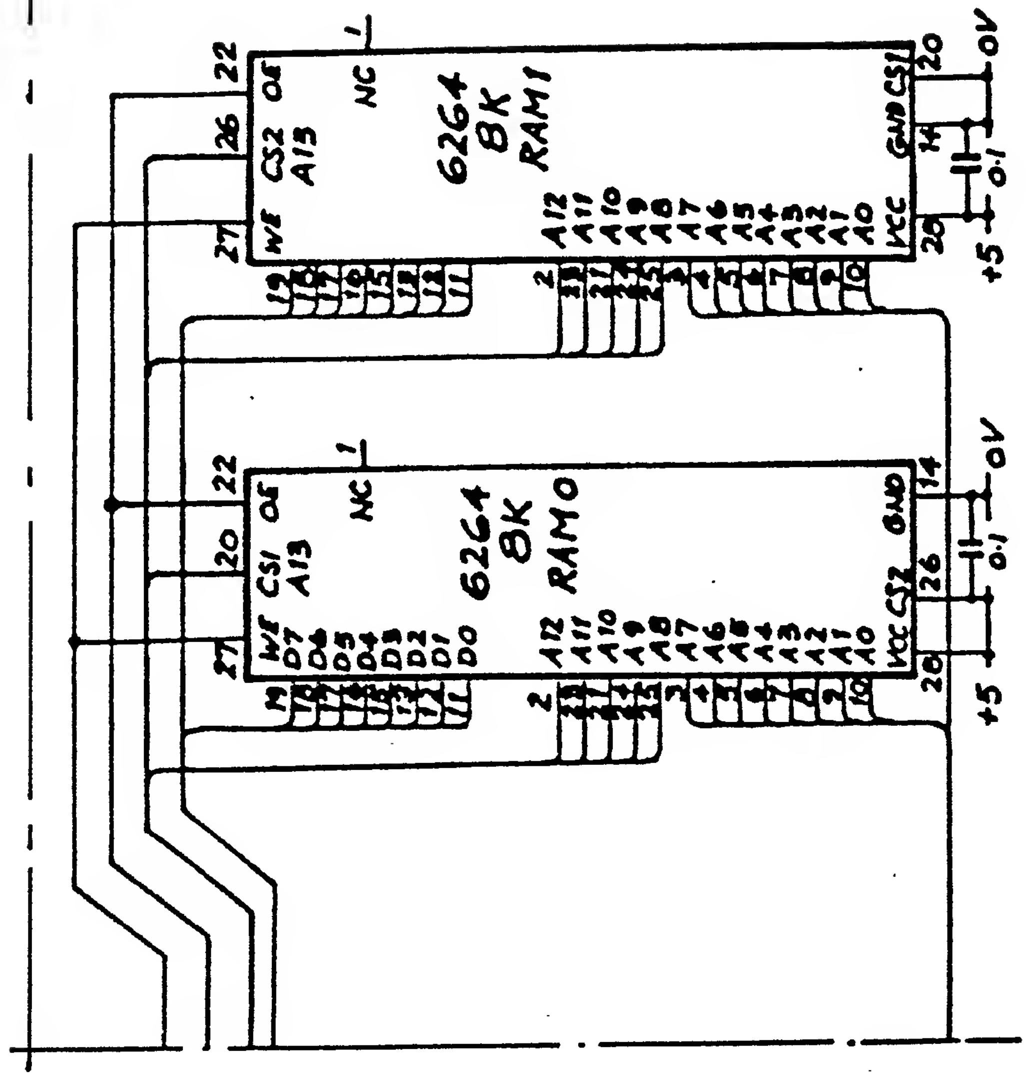


FIG.11b'''

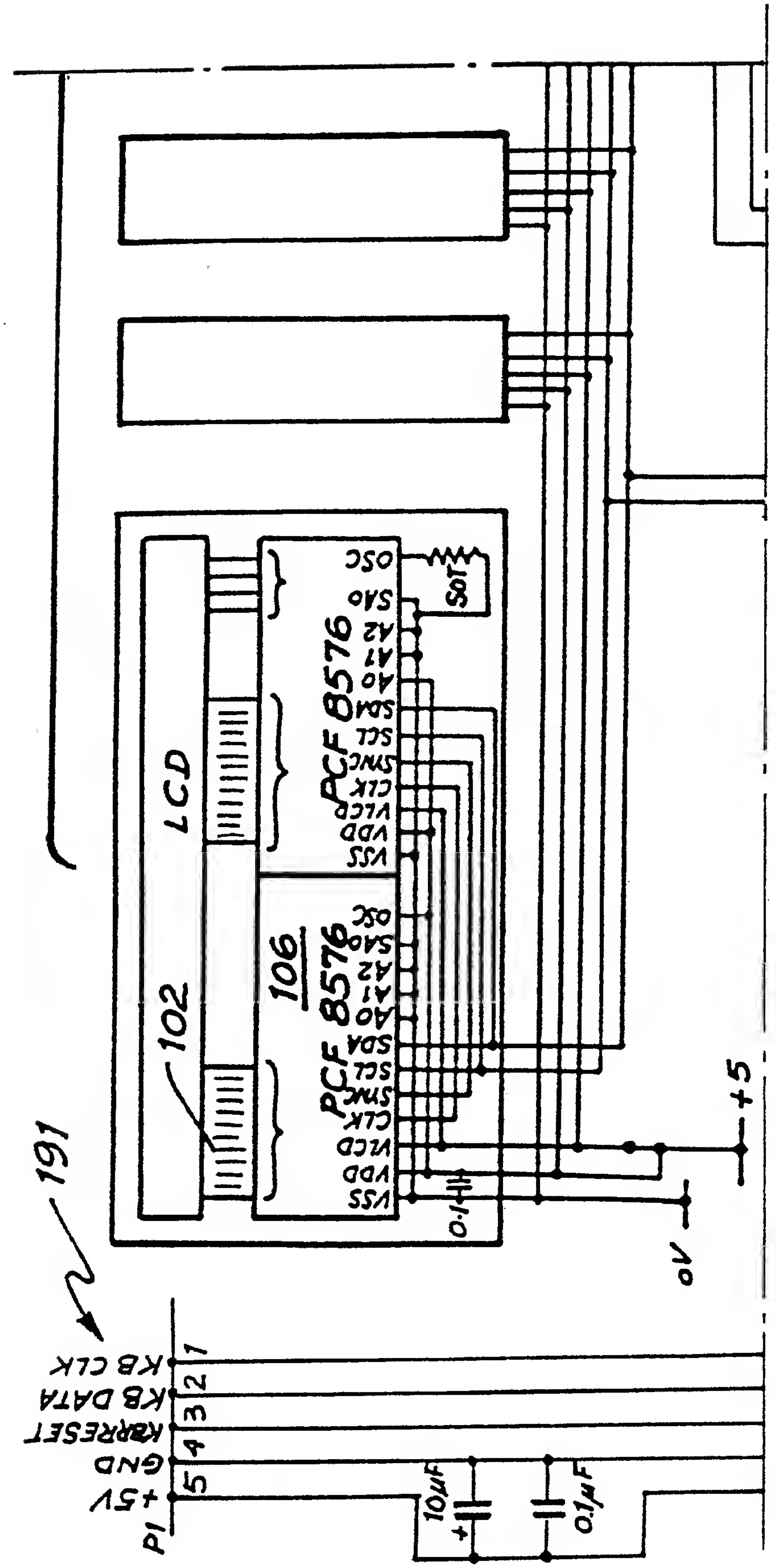

191
LCD
102
106
PCF 8576
PCF 8576
OSC
SAO
A2
A1
A0
SDA
SCL
SYNC
CLK
VLCD
VDD
VSS
0.1
+5
0V
P1
1
2
3
4
5
KB CLK
KB DATA
KBRESET
GND
+5V
10µF
0.1µF

FIG.12a'

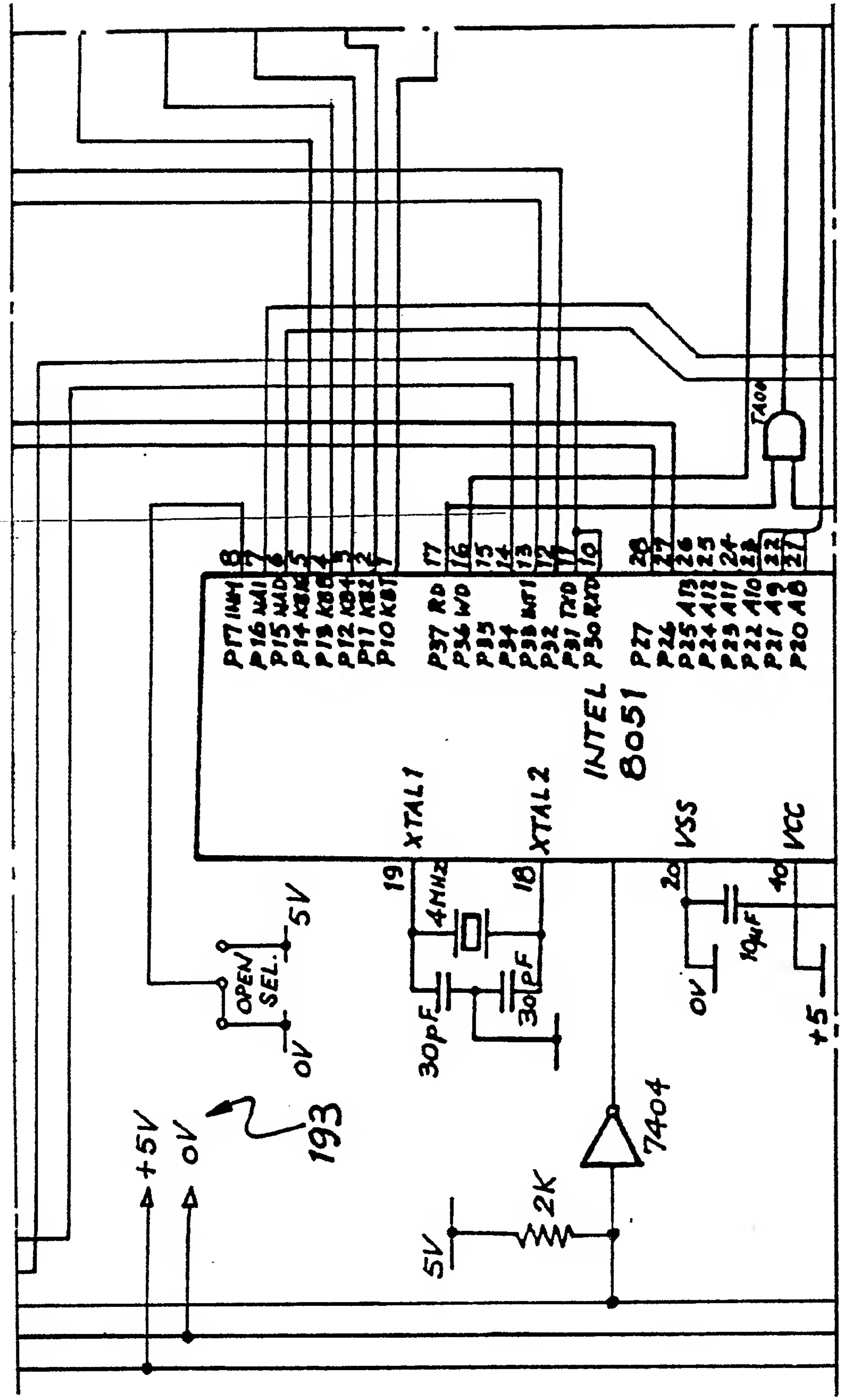

INTEL
8051
XTAL1
XTAL2
VSS
VCC
P17 INH
P16 WAI
P15 WAD
P14 KB8
P13 KB8
P12 KB4
P11 KB2
P10 KB1
P37 RD
P36 WD
P35
P34
P33 INT1
P32
P31 TXD
P30 RXD
P27
P26
P25 A13
P24 A12
P23 A11
P22 A10
P21 A9
P20 A8
4MHz
30pF
10μF
7404
2K
5V
+5V
0V
OPEN
SEL.
193

FIG.12a''

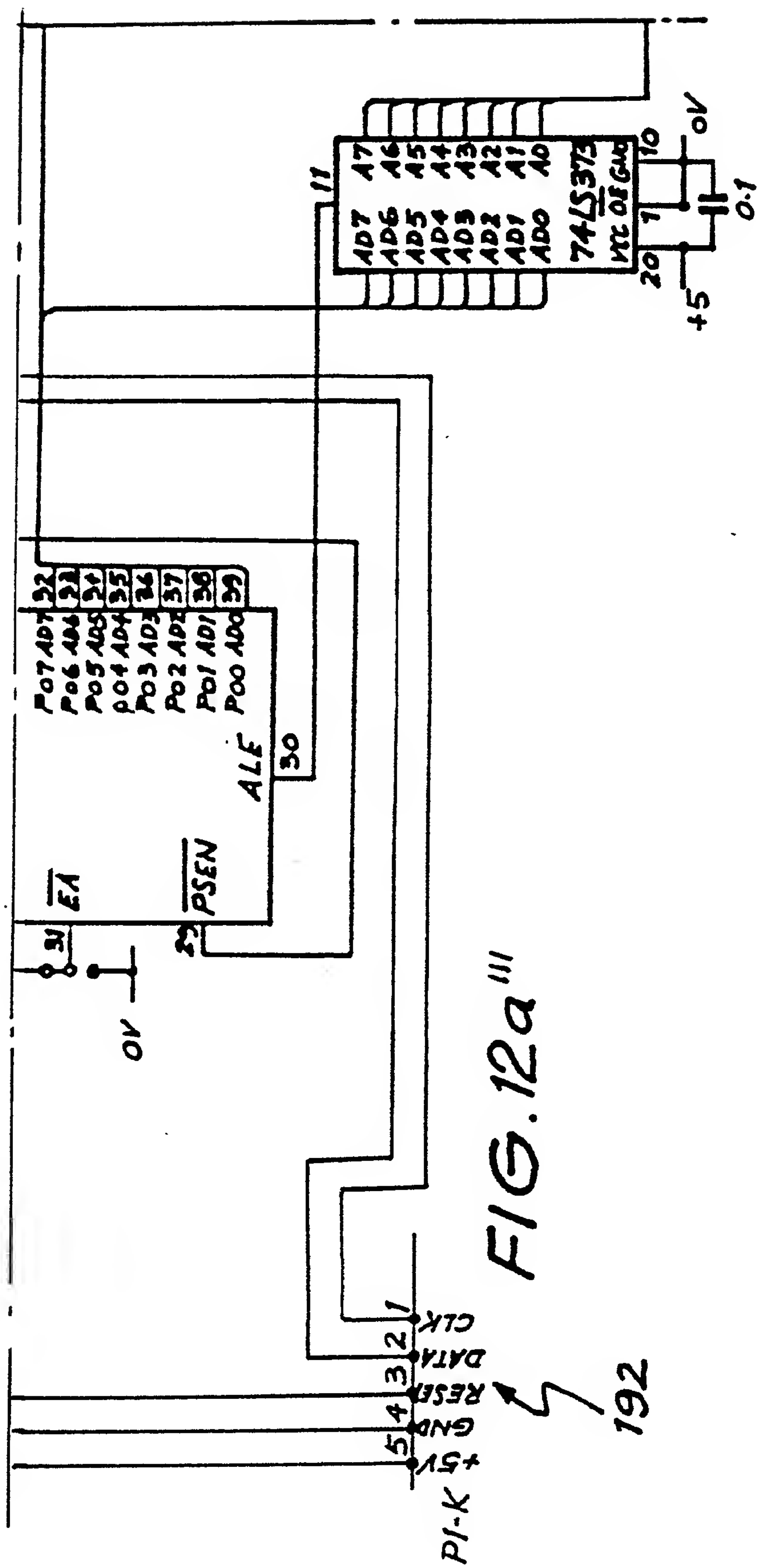


FIG. 12a'''

192

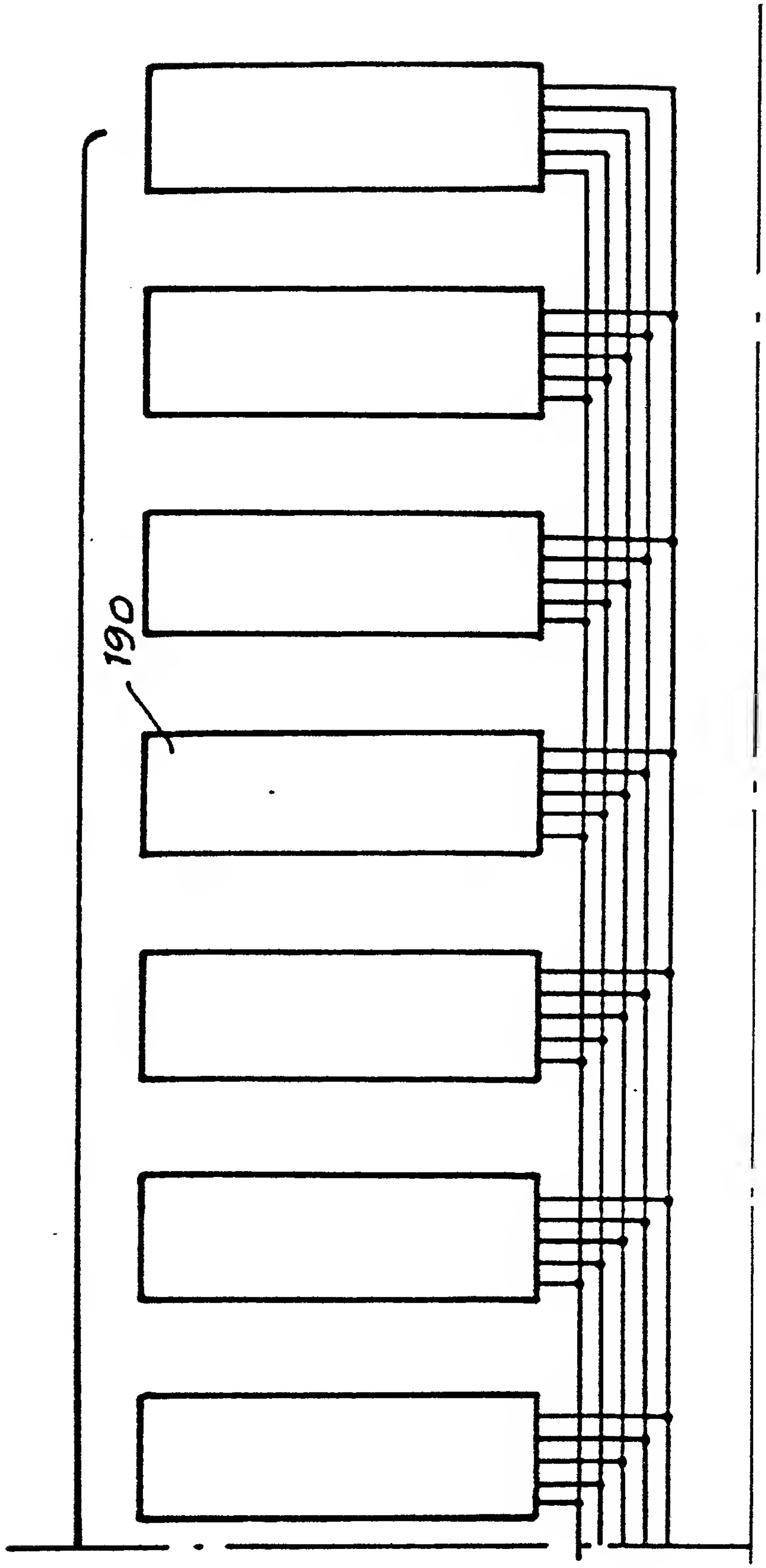


FIG.12b'

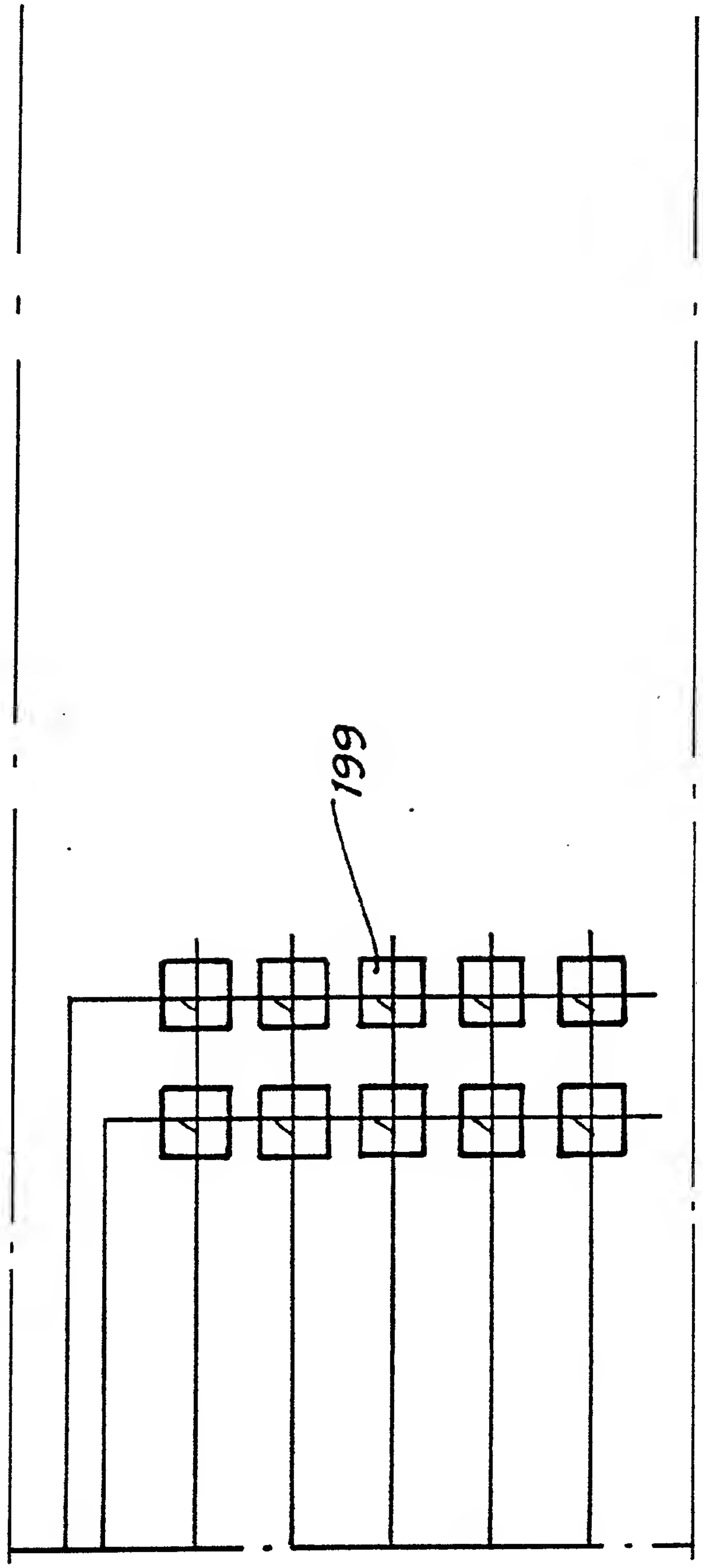


FIG.12b''

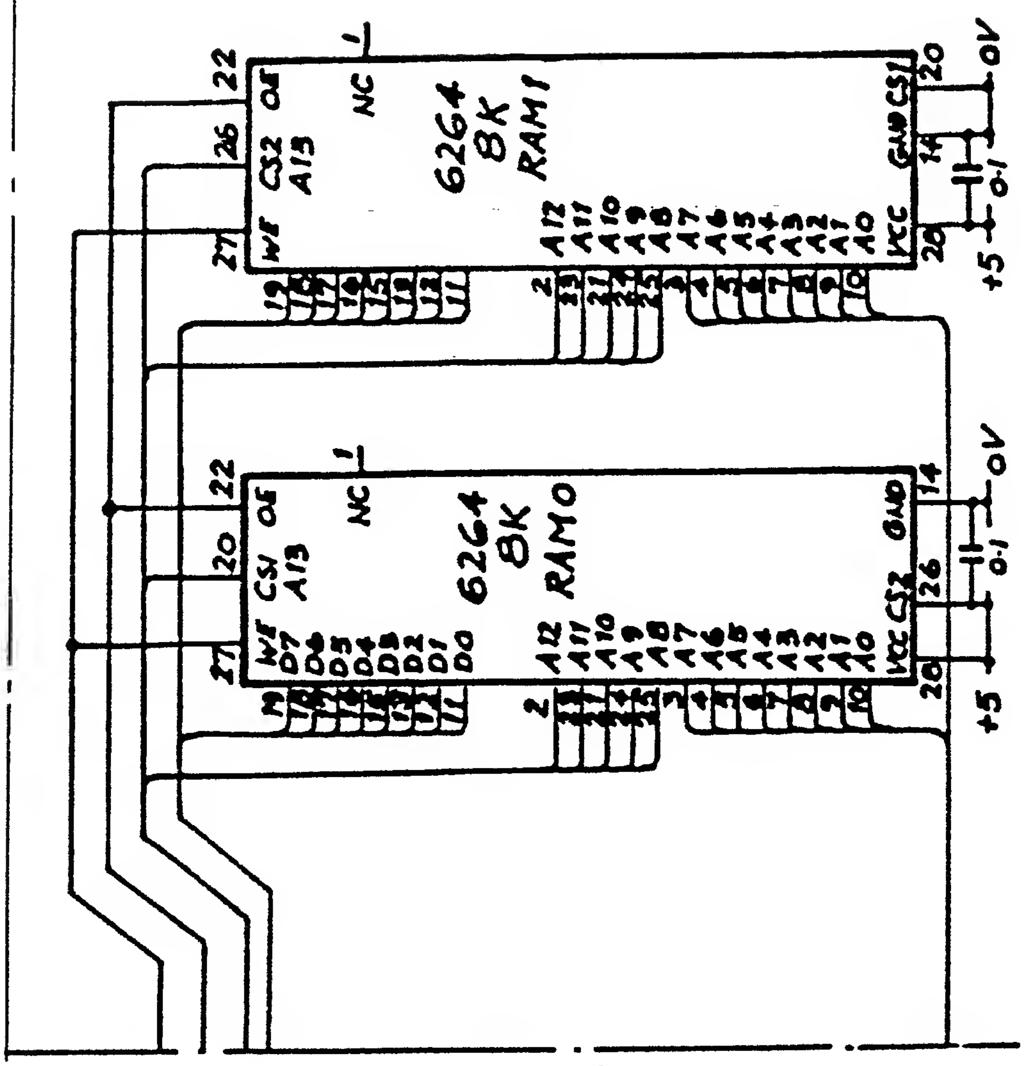


FIG. 12b'''

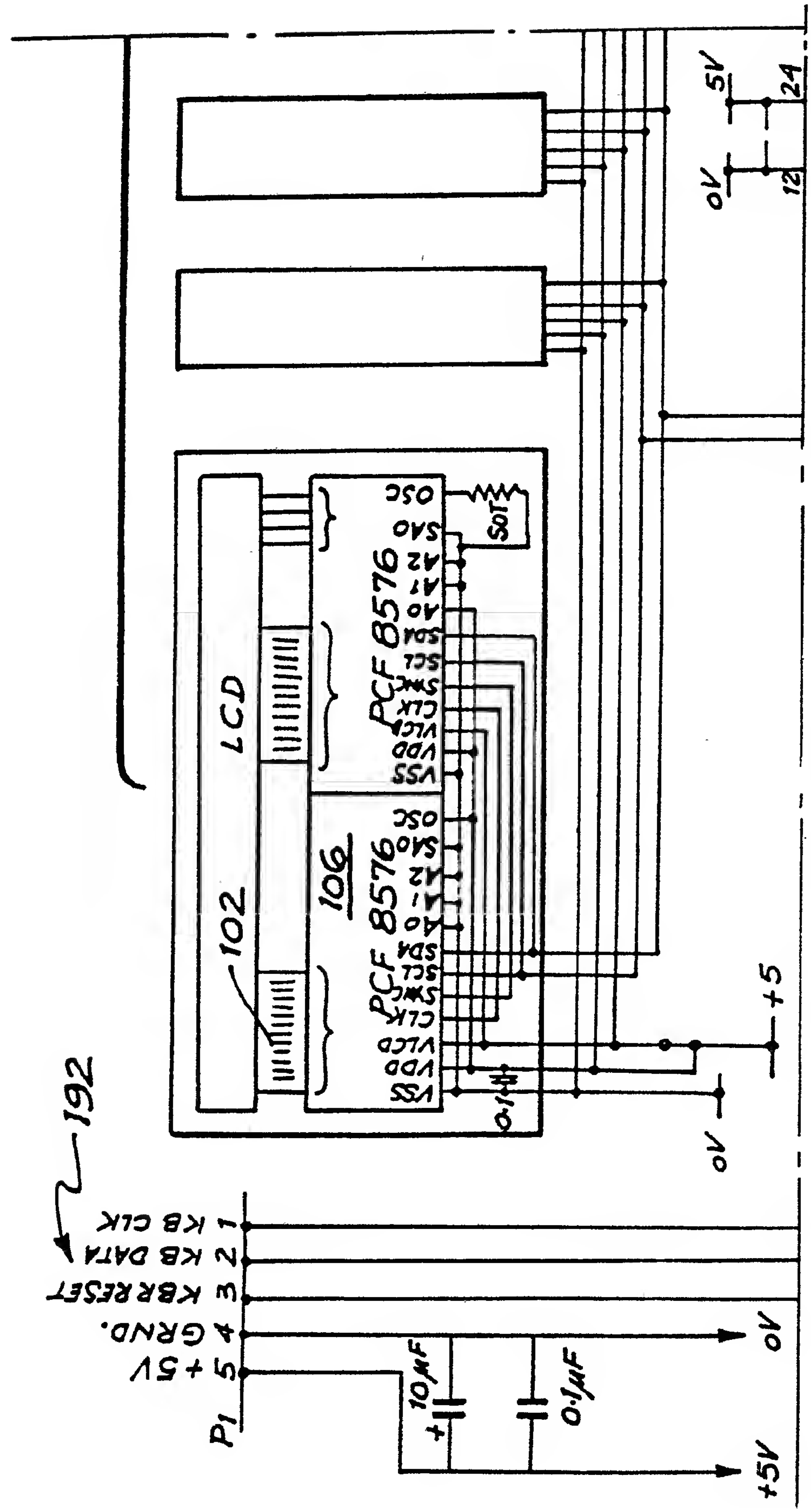

5V
24
0V
12
LCD
102
106
PCF 8576
PCF 8576
OSC
SAO
A2
A1
A0
SDA
SCL
SYNC
CLK
VLCD
VDD
VSS
50T
0.1
+5
0V
192
P1
1 KB CLK
2 KB DATA
3 KBRESET
4 GRND.
5 +5V
10μF
0.1μF
0V
+5V

FIG.13a'

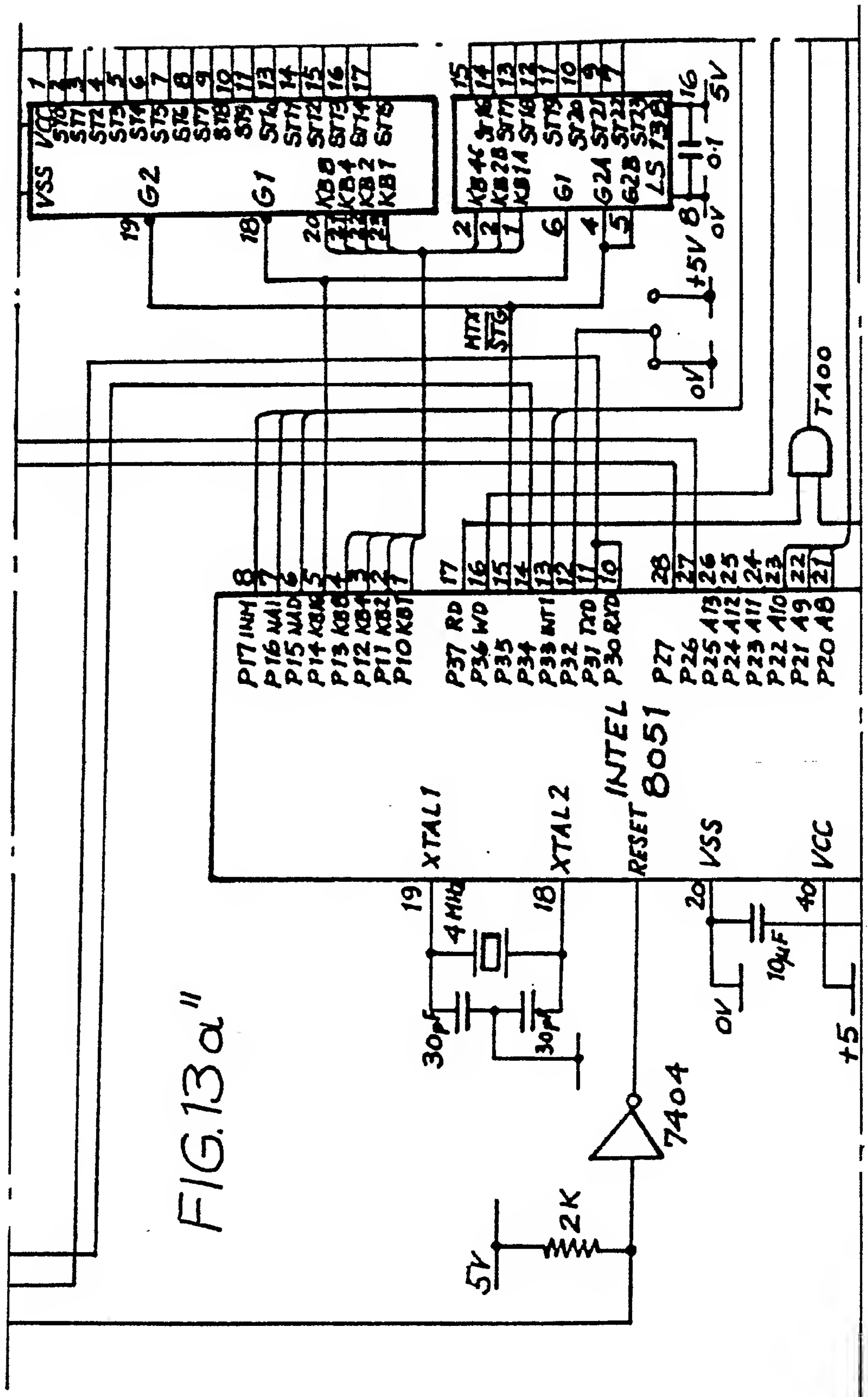

INTEL 8051
XTAL1
XTAL2
RESET
VSS
VCC
4 MHz
30pF
30pF
10μF
7404
2K
5V
+5
0V
TA00
LS 138
G1
G2

FIG.13a''

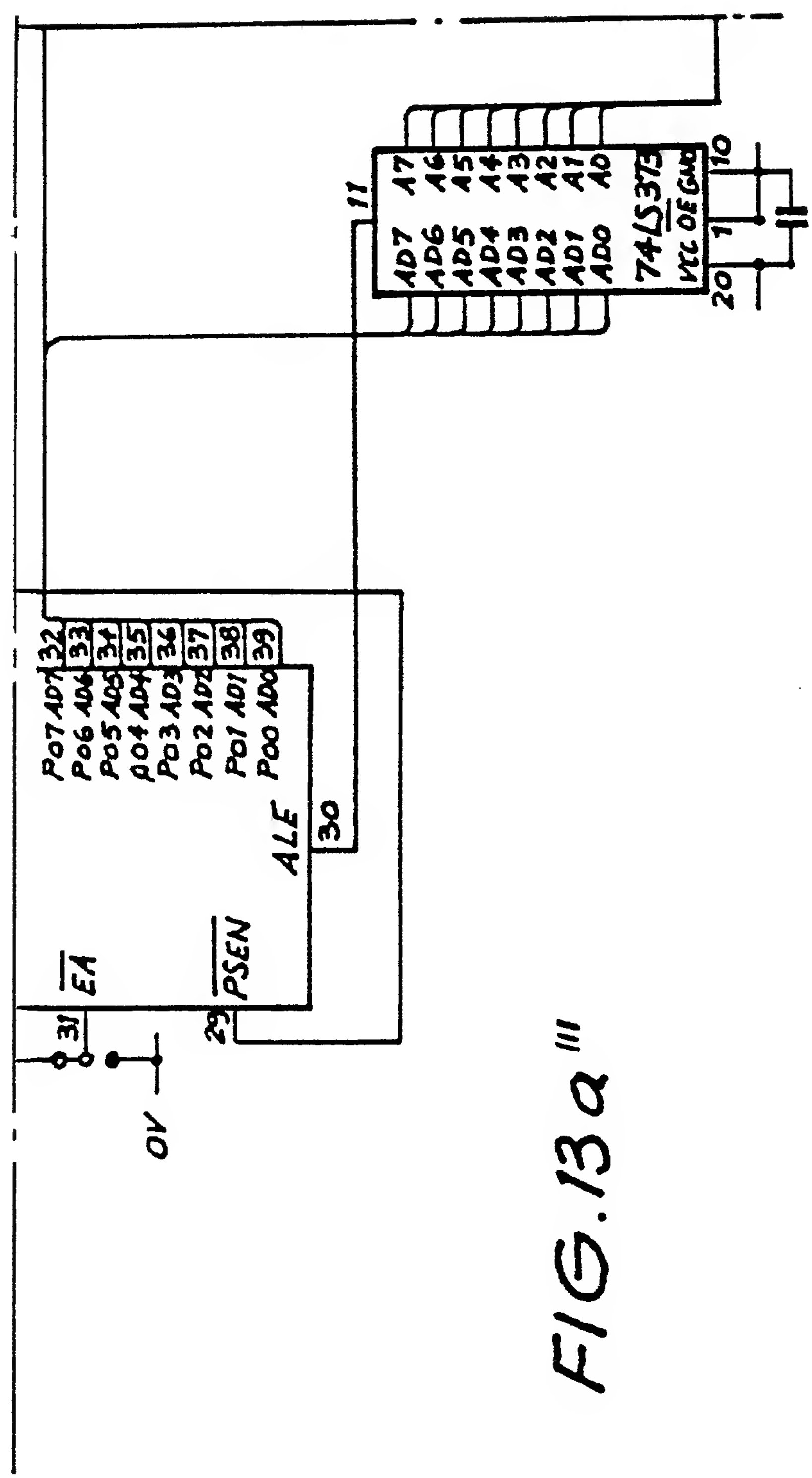

74LS373
AD7 AD6 AD5 AD4 AD3 AD2 AD1 AD0
A7 A6 A5 A4 A3 A2 A1 A0
VCC OE GND
11
20 1 10
P07 AD7 32
P06 AD6 33
P05 AD5 34
P04 AD4 35
P03 AD3 36
P02 AD2 37
P01 AD1 38
P00 AD0 39
ALE 30
EA 31
PSEN 29
0V

FIG. 13a'''

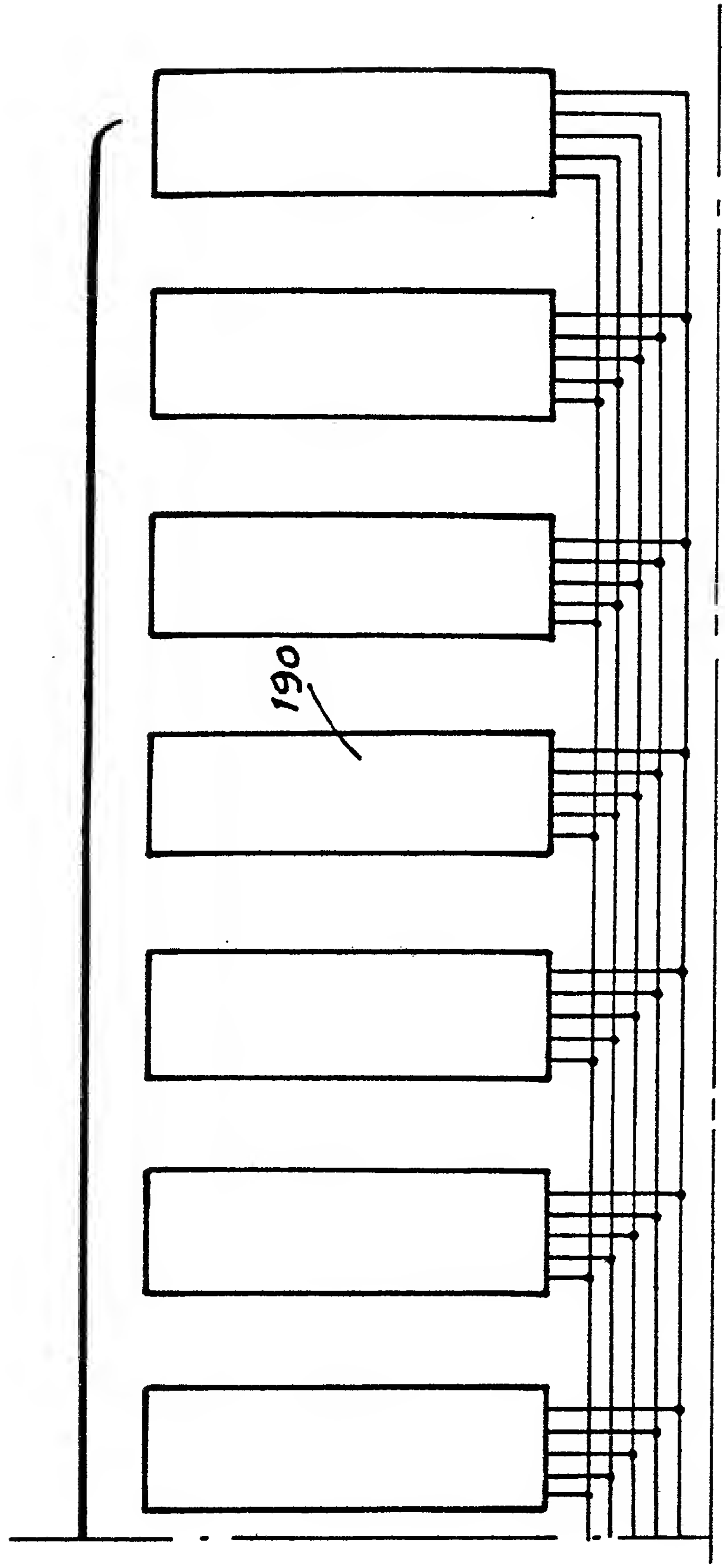


FIG. 13b'

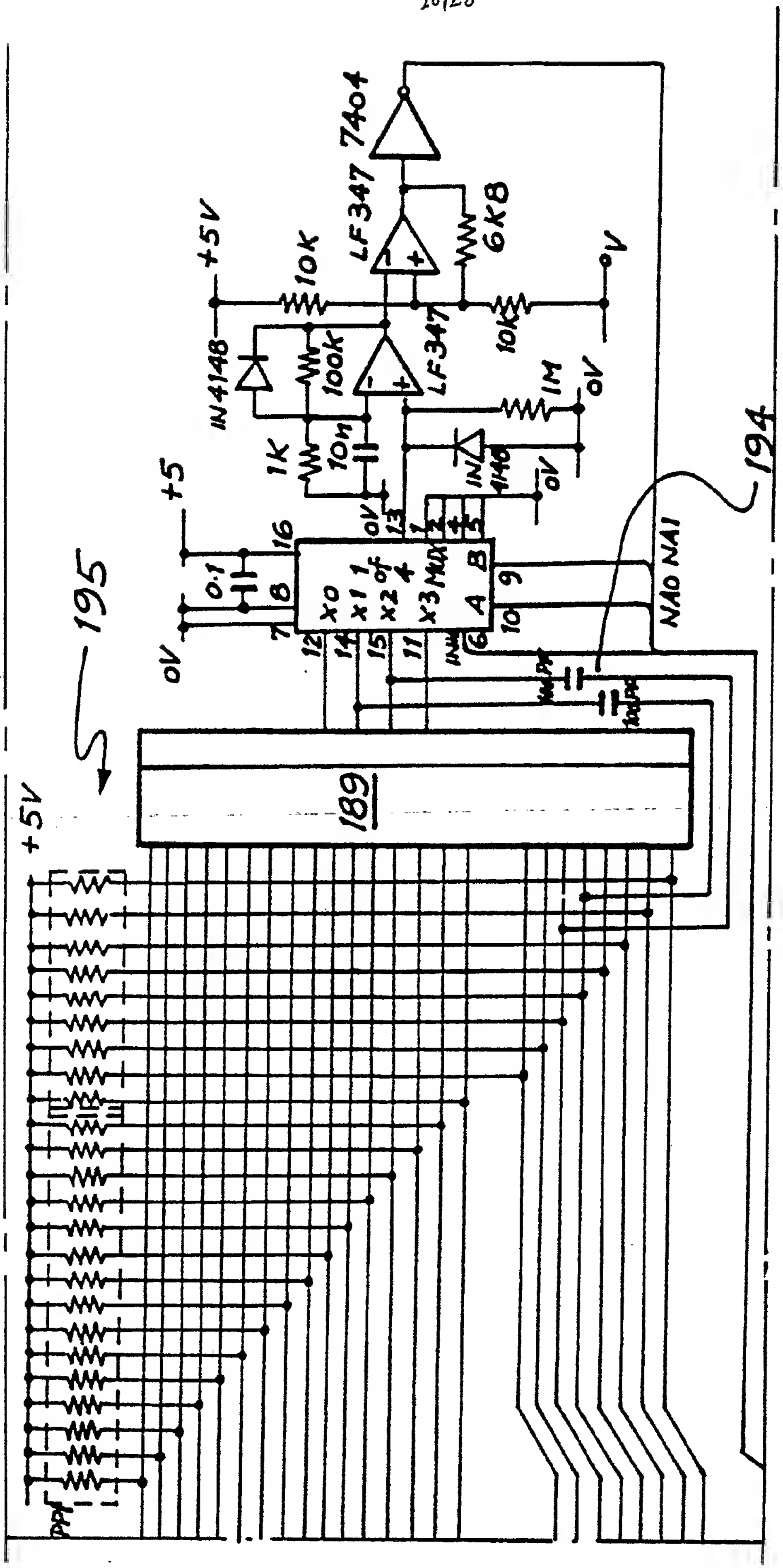

7404
LF347
6K8
+5V
10K
0V
1N4148
100k
LF347
10k
1M
0V
1K
10n
1N4148
0V
+5
16
13
0V
0.1
8
X0
X1
X2
X3
1 of 4 MUX
B
A
9
10
NA0 NA1
194
195
7
12
14
15
11
6
189
+5V

FIG.13b''

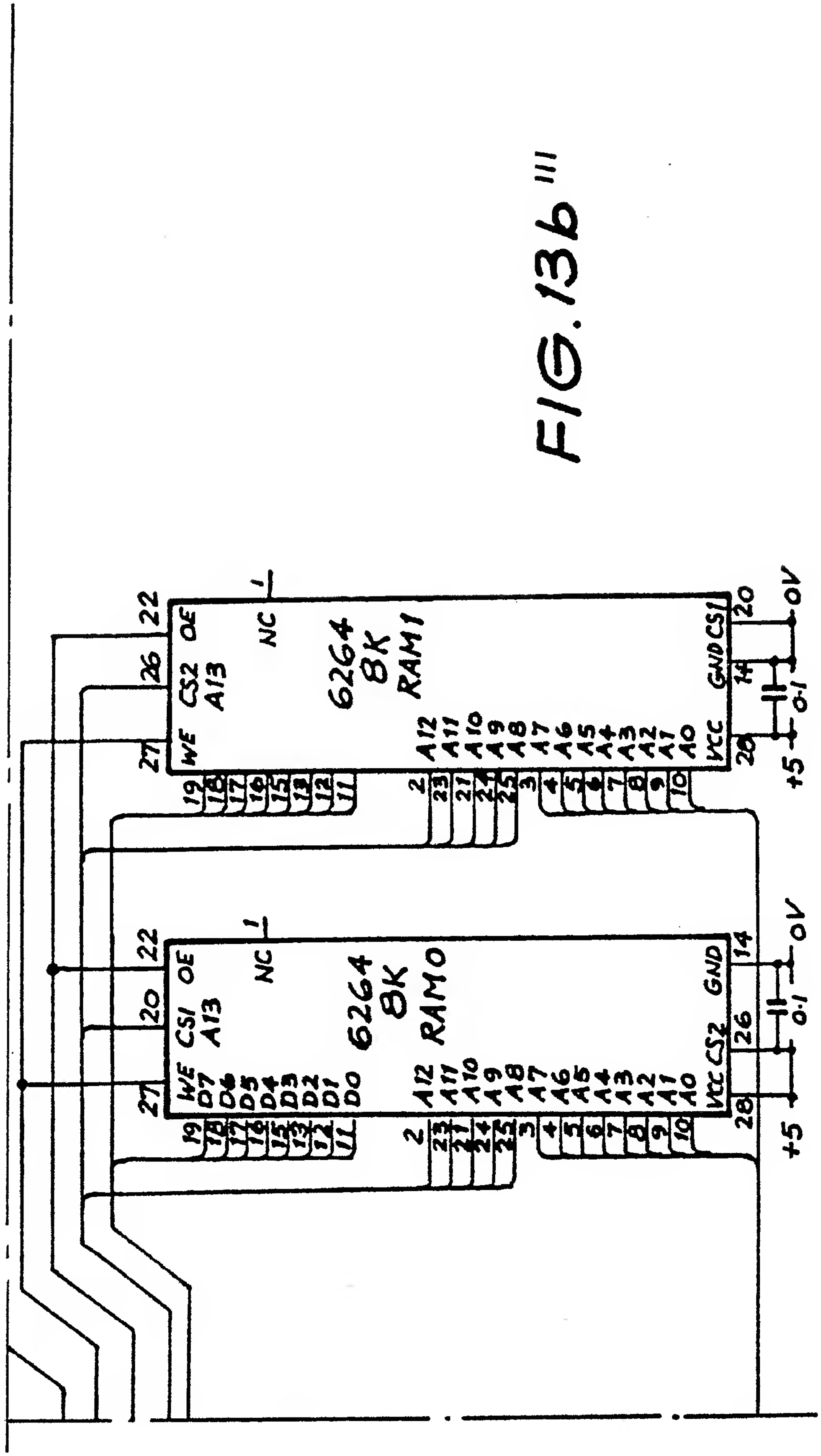


FIG. 13b'''

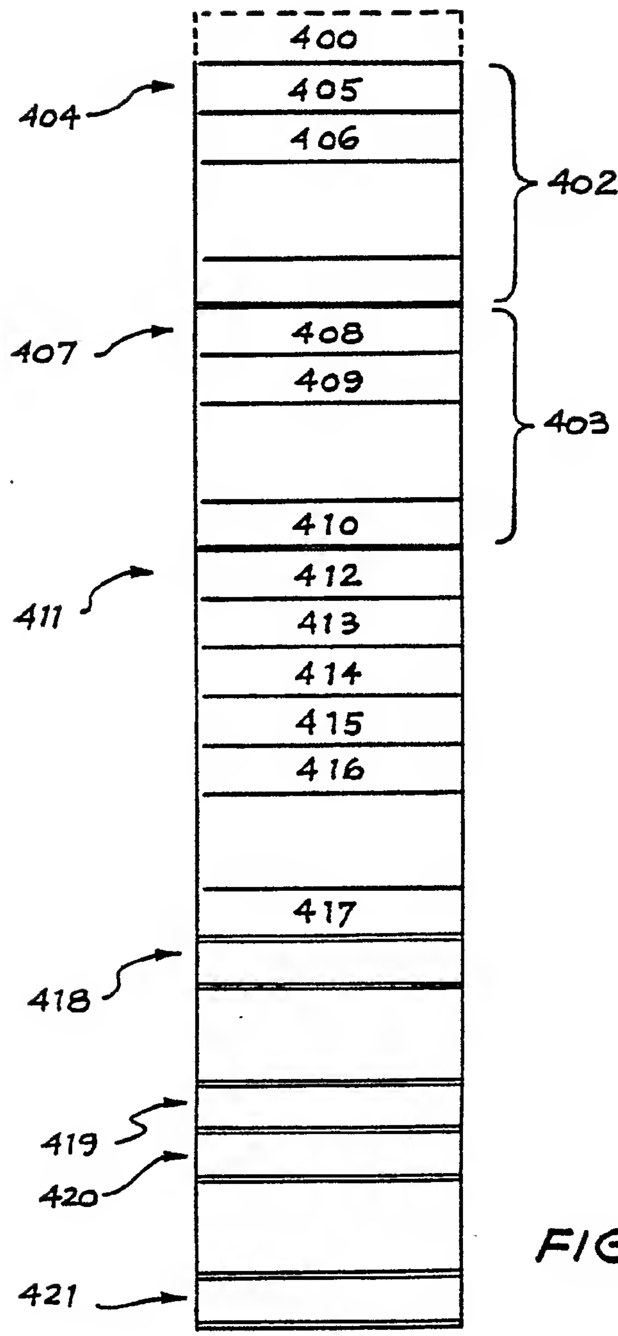

400
405
406
404
402
407
408
409
403
410
411
412
413
414
415
416
417
418
419
420
421

FIG. 14

INTERNATIONAL SEARCH REPORT

International Application No PCT/AU 84/00193

I. CLASSIFICATION OF SUBJECT MATTER (if several classification symbols apply, indicate all) [3]

According to International Patent Classification (IPC) or to both National Classification and IPC

Int. Cl.[3] G06F 3/02

II. FIELDS SEARCHED

Minimum Documentation Searched [4]

Classification System	Classification Symbols
IPC	G06F 3/02

Documentation Searched other than Minimum Documentation to the Extent that such Documents are Included in the Fields Searched [5]

AU: IPC as above

III. DOCUMENTS CONSIDERED TO BE RELEVANT [14]

Category *	Citation of Document, [16] with indication, where appropriate, of the relevant passages [17]	Relevant to Claim No. [18]
X	US, A, 3340401 (XEROX) 26 December 1963 (26.12.63)	(1,3,4,13,14)
X	US, A, 3777222 (IMBC) 12 May 1972 (12.05.72)	(1,3,4,13,14)
X	JP, A, 55-157028 (SUWA SEIKOSHA K.K.) 23 May 1979 (23.05.79) (JAPATIC English Language Abstract)	(1-17)
X	JP, A, 53-123034 (RICOH K.K.) 4 February 1977 (04.02.77) (JAPATIC English Language Abstract)	(1,3,4,13,14)
X	JP, A, 55.41507 (OKI DENKI KOGYO K.K.) 18 September 1978 (18.09.78) (JAPATIC English Language Abstract)	(1-17)

* Special categories of cited documents: [15]

"A" document defining the general state of the art which is not considered to be of particular relevance

"E" earlier document but published on or after the international filing date

"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)

"O" document referring to an oral disclosure, use, exhibition or other means

"P" document published prior to the international filing date but later than the priority date claimed

"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention

"X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step

"Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art.

"&" document member of the same patent family

IV. CERTIFICATION

Date of the Actual Completion of the International Search [2]	Date of Mailing of this International Search Report [2]
17 December 1984 (17.12.84)	19 DECEMBER 1984 (19.12.84)
International Searching Authority [1]	**Signature of Authorized Officer** [20]
Australian Patent Office	W.J. CLARKSON

ANNEX TO THE INTERNATIONAL SEARCH REPORT ON
INTERNATIONAL APPLICATION NO. PCT/AU **84/193**

This Annex lists the known "A" publication level patent family members relating to the patent documents cited in the above-mentioned international search report. The Australian Patent Office is in no way liable for these particulars which are merely given for the purpose of information.

Patent Document Cited in Search Report	Patent Family Members		
US 3777222	DE 2308769 JP 49028232	FR 2185254	GB 1429682

END OF ANNEX